

Федеральное государственное образовательное бюджетное учреждение  
высшего образования  
«Финансовый университет при Правительстве Российской Федерации»

*На правах рукописи*

Тимофеев Александр Николаевич

МОДЕЛЬ И МЕТОДИКА  
ОПЕРАТИВНОГО УПРАВЛЕНИЯ  
ИНДИВИДУАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ  
ТРАЕКТОРИЕЙ ПРИ ЭЛЕКТРОННОМ  
ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

1.2.2. Математическое моделирование, численные методы  
и комплексы программ

ДИССЕРТАЦИЯ  
на соискание ученой степени  
кандидата технических наук

Научный руководитель

Михайлова Светлана Сергеевна,  
доктор экономических наук, доцент

Москва – 2025

## Оглавление

Введение.....	5
Глава 1 Обоснование необходимости совершенствования существующего научно-методического аппарата организации управления процессом электронного обучения.....	20
1.1 Анализ применения технологий искусственного интеллекта в области электронного обучения .....	20
1.2 Научно-методический аппарат организации управления процессом электронного обучения на основе индивидуализации и адаптации .....	34
1.3 Анализ исследований в области адаптивного обучения программированию .....	45
1.4 Постановка задачи исследования.....	50
Глава 2 Модель взаимодействия участников учебного процесса для оперативного формирования ИОТ .....	56
2.1 Обоснование требований к модели взаимодействия участников учебного процесса для оперативного формирования ИОТ в целях достижения заданного уровня ЗУН за регламентированное время .....	56
2.2 Обоснование состава и структуры модели взаимодействия участников учебного процесса .....	59
2.3 Описание частных моделей, входящих в состав модели взаимодействия участников учебного процесса.....	82
Глава 3 Методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию .....	103

3.1 Описание основных действий методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию.....	104
3.2 Формирование индивидуальной образовательной траектории для применения в методике. ....	107
3.3 Распределение времени участников учебного процесса. ....	119
3.4 Построение контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.....	124
<b>Глава 4 Исследование эффективности методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию.....</b>	<b>138</b>
4.1 Постановка задачи вычислительного эксперимента с использованием методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию.....	138
4.2 Практические рекомендации по созданию комплекса программ оперативного формирования индивидуальной образовательной траектории .....	160
<b>Заключение .....</b>	<b>177</b>
Список сокращений и условных обозначений.....	179
Словарь терминов .....	180
Список литературы .....	185
Приложение А Результаты анализа профессиональных стандартов.....	210
Приложение Б Примеры задач для решения.....	216
Приложение В Сценарии взаимодействия .....	223
Приложение Г Пример диалога обучающегося с агентом.....	228

Приложение Д Вычислительный эксперимент. Первая группа.....	231
Приложение Е Вычислительный эксперимент. Вторая группа .....	239
Приложение Ж Результаты опроса студентов и преподавателей .....	247
Приложение И Формирование контекста агента на Python .....	251

## Введение

**Актуальность темы исследования.** Современное образование характеризуется активным внедрением цифровых технологий, что способствует переходу от традиционных форм обучения к электронным. Одной из ключевых тенденций является персонализация образовательного процесса, которая реализуется через построение индивидуальных образовательных траекторий (далее – ИОТ). Несмотря на широкое распространение адаптивных обучающих систем, задача обеспечения достижения обучающимися заданного уровня знаний, умений и навыков (далее – ЗУН) в установленные сроки остаётся недостаточно решенной. В условиях роста сложности образовательных программ и увеличения объёма информации традиционные методы формирования ИОТ сталкиваются с рядом ограничений, среди которых можно выделить отсутствие учета влияния деструктивных факторов, формирование ИОТ без корректировки в реальном времени и высокую трудоёмкость ручного управления процессом обучения. Существующие электронные образовательные системы (далее – ЭОС) не способны оперативно адаптироваться к изменяющимся условиям и индивидуальным особенностям обучающихся, что снижает их эффективность.

Важным направлением развития электронного обучения является применение технологий искусственного интеллекта, в частности, особого вида нейронных сетей глубокого обучения (далее – НГО) – больших языковых моделей (далее – БЯМ) и многоагентных систем. Эти технологии позволяют автоматизировать процессы взаимодействия участников образовательного процесса, обеспечивать персональные рекомендации и динамически корректировать ИОТ. Однако их использование часто ограничивается отсутствием комплексных моделей, учитывающих как индивидуальные, так и групповые аспекты обучения, а также влияние деструктивных факторов на результат. Особую актуальность приобретает задача обучения программированию. В рамках диссертационного исследования автором был

проведен опрос 174 студентов и 31 преподавателя из трех высших учебных заведений и одного среднеспециального учебного заведения, который показал высокий интерес к изучению программирования (58% студентов и преподавателей) и запрос на индивидуализацию этого процесса (68% студентов и 81% преподавателей). Результаты опроса приведены в приложении Ж.

Существующие ЭОС не обеспечивают достижения гарантированного результата в регламентированные сроки, что обусловлено отсутствием учёта деструктивных факторов.

Разработка модели и методики оперативного формирования индивидуальной образовательной траектории, обеспечивающей достижение заданного уровня ЗУН за регламентированное время, является актуальной научной задачей, решение которой позволит повысить эффективность электронного обучения. Предлагаемый в работе подход основан на интеграции нейронных сетей глубокого обучения, многоагентного подхода и марковских процессов принятия решений, что позволяет учесть недетерминированность учебного процесса и влияние деструктивных факторов. Применение алгоритма генерации с дополненной выборкой для формирования базы знаний нейросетевых агентов, позволят создать интеллектуальную образовательную систему, способную адаптироваться к изменяющимся условиям и потребностям обучающихся. Всё вышесказанное свидетельствует о высокой актуальности и практической значимости проводимого исследования, результаты которого могут быть применены не только в сфере обучения программированию, но и в других областях электронного образования.

**Степень разработанности темы исследования.** Вопросам формирования и корректировки индивидуальных образовательных траекторий в условиях электронного обучения, в том числе с применением технологий искусственного интеллекта, посвящено значительное количество исследований. Теоретические и методические основы адаптивного обучения,

моделирования учебного процесса и проектирования интеллектуальных образовательных систем исследовали такие учёные, как А.А. Власенко, М.А. Косоногова, И.А. Кречетов, Е.Ю. Благов, И.А. Лещева, С.А. Щербан, О.Х.З. Аль-Дулаими, Р.Е. Медведев, Ф.Х. Нгуен, F. Osborne, А.А. Сеньковская, А.В. Макаров, О.В. Перченок.

Вопросы применения больших языковых моделей при построении индивидуальной образовательной траектории и адаптивном обучении программированию изучали S. Xu, X. Zhang, L. Qin, B. Radmehr, A. Singla, T. Käser, C. Chen, A. Leitch, J. Lu, J. Chen, C. Lee, J.Xiong, A. Salmon, K. Hammer, P. Denny, V. Kumar, N. Giacaman, C. Kerslake, J. Prather, H. Keuning, N. Kiesler, D. Schiffner, A. Savelka, A. Luxton-Reilly.

Интеграция адаптивного обучения программированию с применением больших языковых моделей в информационные системы представлены в работах C. Qian, P. Denny, T. Wang.

Подходы к реализации многоагентных систем рассматривали В.Б. Тарасов, А.В. Кузнецов, Ю.Ф. Тельнов, В.А. Казаков, Т.М. Шамсутдинова, C.A. Iglesias, M. Garijo, J.C. Gonzalez.

Несмотря на значительный объём исследований в области адаптивного обучения, искусственного интеллекта и управления ИОТ, остаются недостаточно изученными вопросы оперативного формирования ИОТ, обеспечивающего достижение ЗУН в регламентированные сроки в условиях воздействия деструктивных факторов. Не разработаны комплексные модели и методики, интегрирующие многоагентный подход, марковские процессы принятия решений и нейросетевые методы для управления образовательным процессом с учётом реального времени и ресурсов. Требуют дальнейшего исследования методы минимизации влияния деструктивных факторов и обеспечения эффективности формирования ИОТ на основе современных подходов.

**Цель и задачи исследования.** Целью настоящего исследования является разработка метода управления индивидуальной образовательной

траекторией в виде комплекса проблемно-ориентированных программ, обеспечивающих достижение требуемого уровня знаний, умений и навыков за установленное время при обучении программированию в интеллектуальной электронной образовательной системе.

Для достижения поставленной цели потребовалось решение следующих задач:

1) Провести анализ современных моделей, методов и технологий искусственного интеллекта, применяемых для формирования индивидуальных образовательных траекторий в электронном обучении. Выявить деструктивные факторы, влияющие на достижение заданного уровня ЗУН в регламентированные сроки, и проанализировать существующие подходы к их учету.

2) Разработать модель взаимодействия участников учебного процесса для оперативного формирования ИОТ, обеспечивающую достижение заданного уровня ЗУН за регламентированное время, включающую:

- частную модель формирования ИОТ на основе марковского процесса принятия решений, связывающую функцию перехода между этапами обучения с текущим уровнем ЗУН и распределением времени;

- частную модель распределения времени участников учебного процесса, учитывающую влияние деструктивных факторов (ошибок обучающегося и ошибок оценки) на временные затраты;

- частную модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой, направленную на минимизацию ошибок интеллектуальных агентов.

3) Разработать методику оперативного формирования индивидуальной образовательной траектории, формализующую процедуру динамической корректировки ИОТ на основе результатов моделирования взаимодействия участников и оценки текущего состояния процесса обучения.

4) Разработать практические рекомендации по созданию комплекса проблемно-ориентированных программ для оперативной корректировки индивидуальной образовательной траектории с учетом соблюдения заданного уровня знаний, умений и навыков и сроков обучения.

**Объект исследования.** Объектом исследования является процесс оперативного принятия решений по динамической корректировке индивидуальной образовательной траектории, направленный на обеспечение достижения заданного уровня знаний, умений и навыков обучающимся в установленные образовательным стандартом сроки. Исследование фокусируется на установлении и использовании зависимостей между функциями принятия решений, распределением времени участников учебного процесса, текущим уровнем ЗУН и влиянием деструктивных факторов в условиях электронного обучения программированию.

**Предмет исследования.** Предметом исследования являются математические модели, методы и алгоритмы, обеспечивающие формирование и корректировку индивидуальных образовательных траекторий в интеллектуальных электронных образовательных системах (далее – ИЭОС). В работе рассматриваются как детерминированные модели, основанные на аналитических зависимостях (иерархические линейные модели, марковские процессы), так и модели, оперирующие с нечеткими, неполными данными и недетерминированностью учебного процесса (многоагентные системы, нейросетевые модели).

**Область исследования** диссертации соответствует п. 8. «Комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента»; п. 9. «Постановка и проведение численных экспериментов, статистический анализ их результатов, в том числе с применением современных компьютерных технологий (технические науки)» Паспорта научной специальности 1.2.2. Математическое моделирование, численные методы и комплексы программ (технические науки).

**Научная новизна исследования.** На основании проведенного теоретического и экспериментального исследования получены следующие новые научные результаты:

1) Разработана модель взаимодействия участников учебного процесса для формирования индивидуальной образовательной траектории при обучении программированию в интеллектуальной электронной образовательной системе, отличающаяся от известных моделей, предложенных А.А. Власенко [19], И.А. Кречетовым [13], Е.Ю. Благовым, И.А. Лещевой, С.А. Щербан [53], Д.А. Ульяновым [25], О.Х.З. Аль-Дулаими [10], F. Osborne и E. Motta [117], D. Edge [35] тем, что:

- двухуровневая зависимость индивидуальной образовательной траектории от текущих оценок показателей восприятия и реакции обучаемого с учетом действующих деструктивных факторов сформированная по результатам представления участников учебного процесса имитационными моделями с функциями интеллектуальных агентов, контекст которых формируется из базы знаний, содержащей онтологический сегмент с сущностями и отношениями, используемыми в семантической модели заданной предметной области;
- параметры ИОТ, формализуемой марковским процессом принятия решений, учитывают результаты прогноза показателей восприятия и реакции обучаемых на корректировку индивидуального учебного плана по данным сформированной нейросети глубокого обучения;
- учет данных о текущем уровне знаний, умений и навыков при прохождении учебной программы для своевременной корректировки следующего этапа ИОТ осуществляется по результатам выполняемой интеллектуальными агентами оценки текущих параметров траектории, что позволяет сформировать зависимость достижения обучающимся необходимого для получения заданных компетенций уровня ЗУН в установленные образовательным стандартом сроки и на ее основе оперативно

корректировать ИОТ для определения рациональной этапности обучения и необходимого времени.

2) Предложена оригинальная методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию для достижения заданного уровня знаний, умений и навыков за регламентированное время с применением, основанных на нейросетях глубокого обучения интеллектуальных агентов (далее – НГО-агенты) и марковского процесса принятия решений, отличающаяся от аналогичных методик Э.В. Мищенко [70], В.Д. Лиференко [64], Т. Wang [131], Р. Denny [98; 99], М.А. Косоноговой [12] тем, что:

- на основании зависимости результатов участников учебного процесса на индивидуальном и групповом уровнях от влияния деструктивных факторов и учета текущего уровня знаний, умений и навыков определяется оптимальный по критерию достижения заданного уровня ЗУН за оставшееся время следующий этап ИУП, который должен быть включен в ИОТ;

- индивидуальная образовательная траектория оперативно формируется и динамически корректируется с применением марковского процесса принятия решений, где функция полезности вычисляется НГО-агентами на индивидуальном уровне с последующим преобразованием на групповом уровне модели взаимодействия участников учебного процесса;

- НГО-агенты используют адаптированную частную модель построения контекста на основе двухсегментной базы знаний, компетентностный сегмент которой формируется с помощью генерации с дополненной выборкой на основе учебно-методических материалов, что позволяет путем применения методики получить оптимальный уровень ЗУН за регламентированное время путем перераспределения времени участников учебного процесса и оперативного реагирования на влияние деструктивных факторов.

3) Проведенный вычислительный эксперимент по подтверждению эффективности методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию показал, что применение методики позволяет перераспределить заданное время в пользу контактной работы обучающегося с интеллектуальными агентами и снизить зависимость достижения заданного уровня знаний, умений и навыков от влияния деструктивных факторов.

Использование в эксперименте реальных данных, полученных при прохождении курса «Структуры и алгоритмы обработки данных», читаемого студентам направления «Прикладная информатика» Восточно-Сибирского государственного университета технологий и управления, позволило провести верификацию методики.

Разработанная методика может быть использована при создании перспективных ИЭОС, существенно сокращая время прохождения ИУП, что позволяет достигнуть более высокого уровня ЗУН без дополнительных затрат.

Сравнительный анализ показал, что применением НГО-агентов и численного метода построения оптимальной ИОТ, реализованных в комплексе программ позволило снизить влияние деструктивных факторов на прохождение ИОТ и уменьшить фактическое время прохождения в среднем на 10–28%, достигая в отдельных темах 38%.

По результатам вычислительного эксперимента и реализации методики сформированы следующие рекомендации по созданию комплекса программ для исследования эффективности методики оперативного формирования индивидуальной образовательной траектории:

- формировать индивидуальную образовательную траекторию с применением марковского процесса принятия решений, где функция полезности вычисляется НГО-агентами на индивидуальном уровне с последующим преобразованием на групповом уровне модели взаимодействия участников учебного процесса;

– использовать для НГО-агентов двухсегментную базу знаний, формируемую с помощью генерации с дополненной выборкой на основе учебно-методических материалов.

Указанные рекомендации целесообразно использовать при разработке перспективных и модернизации существующих интеллектуальных электронных образовательных систем.

**Теоретическая и практическая значимость работы.** Теоретическая значимость исследования состоит в том, что в отличии от известных моделей формирования ИОТ, ориентированных на статическое планирование или адаптацию на основе начального профиля обучающегося, разработанная модель взаимодействия участников учебного процесса предоставляет теоретический аппарат для динамического управления образовательной траекторией в реальном времени. Это достигается за счет комплексирования методов нейронных сетей глубокого обучения, многоагентного подхода и марковских процессов принятия решений, что позволяет формализовать и учесть влияние деструктивных факторов и недетерминированности учебного процесса.

Разработана оригинальная модель взаимодействия участников учебного процесса, включающая частные модели формирования индивидуальной образовательной траектории, распределения времени участников учебного процесса и построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

Предложена оригинальная методика формирования индивидуальной образовательной траектории при электронном обучении программированию.

Разработанные модели и методика могут служить базой для постановки и решения более сложных задач в области адаптивного обучения, таких как оптимизация распределения ресурсов в ИЭОС, прогнозирование академической успеваемости.

Практическая значимость работы состоит в готовности методики оперативного формирования ИОТ к внедрению в современные электронные образовательные системы на основе данных в главе 4 рекомендаций. Разработанный подход позволяет повысить эффективность обучения программированию за счет достижения целевых показателей ЗУН в регламентированные сроки. Результаты вычислительного эксперимента демонстрируют возможность перераспределения времени прохождения ИОТ в пользу контактной работы обучающихся с интеллектуальными агентами с одновременным достижением заданного уровня ЗУН.

**Методология и методы исследования.** Методологическую основу исследования составили фундаментальные работы российских и зарубежных ученых в области теории адаптивного обучения, искусственного интеллекта в образовании, многокритериального принятия решений и моделирования сложных социально-педагогических систем. Исследование опирается на системный подход, объединяющий методы математического моделирования, многоагентные системы и марковские процессы принятия решений. Для формализации учебного процесса применялись аппарат теории множеств и теория графов, позволившие представить индивидуальный учебный план как ориентированный граф состояний. Обработка естественного языка и генерация образовательного контента осуществлялись с использованием методов глубокого обучения и алгоритма генерации с дополненной выборкой. Практическая реализация моделей и алгоритмов выполнена с применением объектно-ориентированного и функционального программирования.

Информационную базу исследования составили: учебно-методические материалы курсов по программированию, данные образовательных стандартов и профессиональных компетенций; результаты исследований факторов эффективности обучения; данные электронных систем обучения о времени прохождения этапов и академических результатах обучающихся; открытые базы знаний (например, DBpedia) и специализированные онтологии предметной области; научные публикации в рецензируемых отечественных и

зарубежных журналах, материалы конференций по искусственному интеллекту в образовании и адаптивным обучающим системам.

**Положения, выносимые на защиту.** В соответствии с поставленными задачами, положения, выносимые на защиту, включают в себя следующее:

1) модель взаимодействия участников учебного процесса для оперативного формирования индивидуальной образовательной траектории, отличающаяся от известных моделей комплексированием двухуровневой зависимости ИОТ от текущих оценок показателей восприятия и действующих деструктивных факторов и нейросетей глубокого обучения, используемых при формировании ИОТ (С. 56-102);

2) методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию для достижения заданного уровня ЗУН за регламентированное время с применением нейросетей глубокого обучения, отличающаяся от известных методик тем, что на основании зависимости результатов участников учебного процесса на индивидуальном и групповом уровнях от действия деструктивных факторов с учетом текущего уровня ЗУН интеллектуальные агенты динамически корректируют ИОТ с применением двухсегментной базы знаний, которая формируется с помощью генерации с дополненной выборкой на основе учебно-методических материалов (С. 103-137);

3) практические рекомендации по реализации комплекса программ для внедрения предложенной методики в интеллектуальные электронные образовательные системы, включая архитектурные решения и методы интеграции, обеспечивающие возможность оперативного формирования и корректировки индивидуальных образовательных траекторий в реальном времени (С. 160-176).

**Степень достоверности, апробация и внедрение результатов исследования.** Достоверность положений исследования подтверждается их апробацией в установленном порядке, применением качественных и количественных методов оценки методов оценки рассматриваемых проблем,

изучением научных публикаций отечественных и зарубежных авторов по тематике исследования.

Основные положения и результаты исследования апробированы на следующих научных конференциях: на Международной научно-практической конференции «Моделирование современных информационных систем в условиях цифровой трансформации» (Санкт-Петербург, Петербургский государственный университет путей сообщения Императора Александра I, 13-14 апреля 2023 г.); на Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет 21 июня 2023 г.); на XVI Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2023) (Москва, ИПУ РАН, 26-28 сентября 2023 г.); на Всероссийской научно-методической конференции с международным участием «Инновационные технологии обучения в вузе в условиях цифровизации и реформирования высшего образования» (г. Улан-Удэ, Восточно-Сибирский государственный университет технологий и управления, 14-16 февраля 2024 г.); на Национальной научно-практической конференции «Образование и наука» (г. Улан-Удэ, Восточно-Сибирский государственный университет технологий и управления, 8–12 апреля 2024 г.); на Всероссийской научно-методической конференции «Трансформация профессионального образования и инновационная деятельность: содержание, технологии, качество» (г. Улан-Удэ, Восточно-Сибирский государственный университет технологий и управления, 19–21 февраля 2025 г.); на Национальной научно-практической конференции «Образование и наука» (г. Улан-Удэ, Восточно-Сибирский государственный университет технологий и управления, 7-11 апреля 2025 г.); на XVIII Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2025) (Москва, ИПУ РАН, 24-26 сентября 2025 г.).

Программа для ЭВМ «Программный модуль «Многоагентная система обучения программированию на основе больших языковых моделей»,

подготовленная по результатам исследования, внесена в Реестр программ для ЭВМ Роспатента (свидетельство о регистрации Роспатента № RU 2025610558 от 13.01.2025; автор: Тимофеев А.Н.; правообладатель: ООО «СибДиджитал»).

Разработанные в диссертации модель и методика были использованы при обучении сотрудников компании ООО «Розенблatt» с применением ПЭВМ «Многоагентная система обучения программированию на основе больших языковых моделей». Применение указанной ПЭВМ позволило повысить эффективность обучения на 15%.

Материалы диссертации используются в практической деятельности федерального государственного бюджетного образовательного учреждения высшего образования «Восточно-Сибирский государственный университет технологий и управления», в части разработанной методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию, исходя из соблюдения заданного уровня знаний, умений и навыков и сроков обучения. По материалам исследования внедрена разработанная в диссертации модель взаимодействия участников учебного процесса для оперативного формирования ИОТ, ожидаемый эффект от внедрения которой заключается в уменьшении трудозатрат преподавателей на 10%, что существенно повысит эффективность обучения. Используются описанные в исследовании частная модель формирования индивидуальной образовательной траектории, частная модель распределения времени участников учебного процесса и частная модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

Материалы диссертации используются в практической деятельности ООО «СибДиджитал». С использованием разработанной в диссертации методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию проведено обучение сотрудников. Эффект от использования методики выразился в ускорении обучения на 12% в связи с уменьшением трудозатрат на

контактную работу и снижем действия деструктивных факторов. Для снижения действия деструктивных факторов корпоративная база знаний была использована как компетентностный сегмент базы знаний интеллектуальных агентов, входящей в базовые компоненты модели взаимодействия участников учебного процесса и используемой в частной модели построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

Результаты исследования используются Кафедрой радиотехнического вооружения ФГКВОУ ВО «Ярославское высшее военное училище противовоздушной обороны имени Маршала Советского Союза Л.А. Говорова», в частности: модель взаимодействия участников учебного процесса для формирования индивидуальной образовательной траектории при обучению программированию в индивидуальной образовательной среде; методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию. Полученные результаты исследования апробированы и использованы при организации проведения практики «Научно-исследовательская работа» с курсантами 4, 5 курсов.

Апробация и внедрение результатов исследования подтверждены соответствующими документами.

**Публикации.** Основные положения и результаты исследования отражены в 7 публикациях общим объемом 6,25 п.л. (авторский объем – 3,785 п.л.), в том числе 5 работ общим объемом 5,11 п.л. (авторский объем – 2,995 п.л.) опубликованы в рецензируемых научных изданиях, определенных ВАК при Минобрнауки России, а также 2 работы общим объемом 1,14 п.л. (авторский объем – 0,79 п.л.) опубликованы в изданиях, индексируемых в международной цитатно-аналитической базе «Scopus».

**Личный вклад автора.** В работах [80; 83-86; 136-137] автору принадлежит постановка задачи, создание математических моделей, разработка комплекса программ, проведение вычислительных экспериментов.

**Структура и объем диссертации.** Общая структура определена целью и задачами исследования. Диссертация состоит из введения, четырех глав, заключения, списка сокращений и условных обозначений, словаря терминов, списка литературы, состоящего из 137 наименований и 8 приложений. Текст диссертации изложен на 253 страницах, содержит 23 таблицы, 39 рисунков, 112 формул.

## Глава 1

### **Обоснование необходимости совершенствования существующего научно-методического аппарата организации управления процессом электронного обучения**

#### **1.1 Анализ применения технологий искусственного интеллекта в области электронного обучения**

**Основные понятия в электронном обучении.** В связи с интенсивным развитием информационных технологий, глобальной сети Интернет и искусственного интеллекта консервативные формы обучения быстро устаревают. Эффективность применения цифровых технологий отмечается преподавателями по всему миру [1]. Особенно заметен этот процесс в сфере компьютерных наук. Обучение программированию с использованием бумажных учебно-методических материалов, доски и мела проигрывает по эффективности компьютерным системам с электронными учебниками, автоматической проверкой программного кода и совместным доступом к его редактированию.

Электронное обучение получает все более широкое развитие в силу эффективности и удобства для участников образовательного процесса. В настоящей работе термин «электронное обучение» определяется как организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников [6].

Согласно п. 1 ст. 16 273-ФЗ «Об образовании в Российской Федерации» электронное обучение это – «организация образовательной

деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий». Под электронной образовательной системой понимается автоматизированная система, обеспечивающая процесс электронного обучения.

Применение ЭОС снижает трудозатраты преподавателей, упрощает доступ обучающихся к учебно-методическим материалам, повышает адаптивность процесса обучения и индивидуализирует его. Однако в условиях увеличения сложности образовательных программ и подразделения современного знания на все большее количество специализированных отраслей, изучение которых требует постоянной адаптации учебных программ и повышения квалификации педагога, необходим поиск новых способов обучения в рамках ЭОС.

Одним из таких способов является адаптивное обучение, которое в настоящей работе рассматривается как процесс целенаправленного приспособления систем ко всему разнообразию условий их функционирования, как внешних, так и внутренних (по отношению к системе), а также активное освоение окружающей среды, а способность к адаптации – адаптивность – как всеобщее и необходимое свойство адаптивных систем [78]. Тогда адаптивное обучение можно рассматривать как разновидность обучения, при котором для обучающегося формируется индивидуальный учебный план, согласно 273-ФЗ «Об образовании в Российской Федерации» обеспечивающий освоение образовательной программы на основе индивидуализации ее содержания с учетом особенностей и образовательных потребностей конкретного обучающегося [6]. Каждая учебная задача ИУП характеризуется тем, что имеет заданные сроки и уровень ЗУН, от которых зависит овладение компетенциями в сроки, установленные учебной программой всего курса.

При прохождении ИУП с учетом непрерывного анализа персональных учебно-значимых характеристик обучающегося [8] формируется

индивидуальная образовательная траектория. В настоящей работе под ИОТ понимается последовательность учебных задач и материалов, которая формируется при прохождении ИУП исходя из анализа персональных характеристик обучающегося и целей обучения [8]. Соотнося понятия ИУП и ИОТ, можно процитировать А. В. Хуторского: «ИОТ – это не индивидуальная программа. Траектория – след от движения. Программа – ее план» [29; 62]. Каждая пройденная учебная задача ИОТ характеризуется тем, что имеет фактические сроки и уровень ЗУН, которые определяют полученные компетенции.

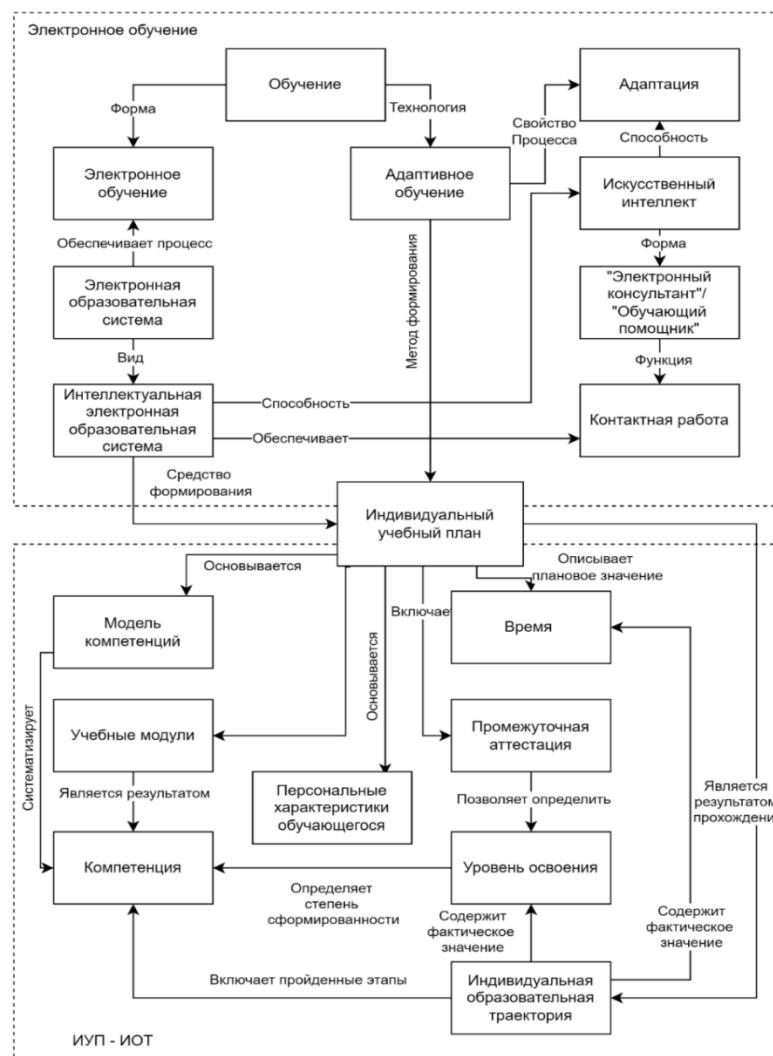
Из определения ИОТ можно сделать вывод о том, что ее формирование и обработка требуют существенных трудозатрат, а эффективная работа с ней в основном возможна только с применением ЭОС, обладающих рядом базовых для адаптивного обучения элементов: возможности выдачи рекомендаций, которые дают обучающимся ранжированные предложения; предиктивной аналитики, которая предсказывает такие метрики, как степень и вероятность достижения целей обучения; единой истории обучения [6].

Перечисленные выше элементы часто характерны для систем, использующих искусственный интеллект. В настоящей работе под искусственным интеллектом (далее – ИИ) понимается способность спроектированной системы приобретать, обрабатывать и применять знания и навыки, а под системой искусственного интеллекта – техническая система, использующая одну или несколько моделей ИИ, которая порождает такие конечные результаты, как контент, прогнозы, рекомендации или решения для заданного набора определенных человеком целей [9].

В контексте адаптивного обучения в ЭОС такие методы ИИ как машинное обучение и инженерия знаний являются основной для «проектирования индивидуальных образовательных маршрутов», «адаптивных систем обучения, автоматически настраивающихся на индивидуальные образовательные маршруты обучающегося и учитывать другие особенности конкретного обучающегося», «конструирования модели

электронных консультантов в помощь обучающимся при выполнении заданий» [7]. Если ЭОС обладает признаками искусственного интеллекта, то будем называть ее интеллектуальной электронной образовательной системой. В зависимости от степени применения ИИ могут быть выделены этапы формирования ИЭОС [77].

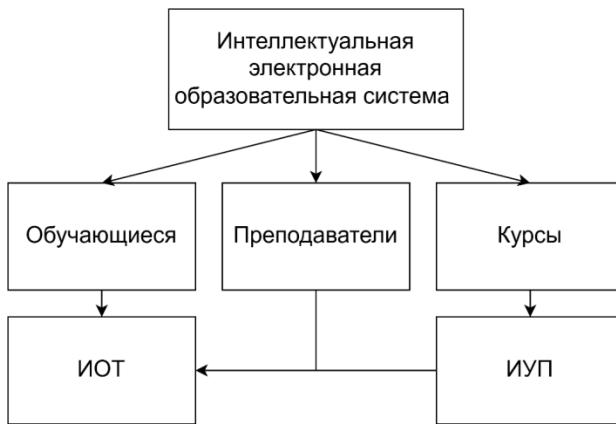
Анализ сферы электронного обучения программированию показал, что ее можно представить в виде структурно-логической схемы, приведенной на рисунке 1, где показано, что адаптивное обучение как метод формирования ИОТ применяется в ЭОС, обладающих «рядом базовых элементов» [7].



Источник: составлено автором по материалам [4; 5; 6; 7; 8; 9].  
Рисунок 1 – Структурно-логическая схема электронного обучения

Современные ИЭОС применяют технологии искусственного интеллекта для эффективной работы с ИОТ в качестве способа адаптивного

обучения. Применение технологий искусственного интеллекта позволяет снизить трудозатраты на учебный процесс. Снижение трудозатрат и повышение адаптивности достигается в том числе проведением части контактной работы с помощью искусственного интеллекта. Структурная схема, соответствующая большинству современных ИЭОС, приведена на рисунке 2.



Источник: составлено автором по материалам [4; 5; 6; 7; 8; 9].

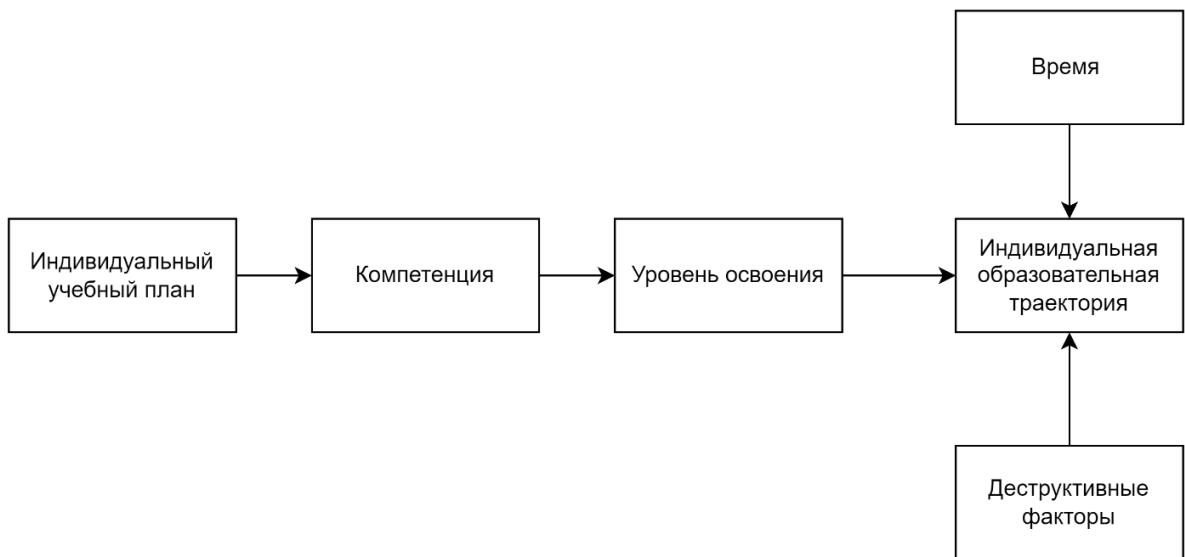
Рисунок 2 – Структурная схема ИЭОС

Эффективная работа с ИОТ обуславливается не только интеллектуализацией ИЭОС, но и учетом ряда факторов, поскольку ИОТ основывается на моделях компетенций, описанных в том числе в профессиональных стандартах, и состоит из учебных модулей с заданной трудоёмкостью и последовательностью, а также средств проверки уровня ЗУН.

Структурно-логическая схема на рисунке 3 показывает, что адаптивное обучение в ИЭОС с применением искусственного интеллекта направлено на достижение заданного уровня ЗУН при электронном обучении программированию в установленные сроки с соблюдением предусмотренных образовательным стандартом требований.

Учебный процесс подвержен влиянию различных деструктивных факторов, например таких как отсутствие обратной связи, ошибки в оценках, запутанное изложение материала и так далее. Текущий методический аппарат

частично учитывает, но не детализирует влияние этих факторов на принятие решений о построении сценария прохождения ИОТ факторов, что показано на рисунке 3.



Источник: составлено автором по материалам [4; 5; 6; 7; 8; 9].

Рисунок 3 – Структурно-логическая схема зависимостей, влияющих на ИОТ

Для освоения компетенций, предусмотренных учебной программой курса, требуется постоянный контроль текущего уровня освоения, соблюдения временных параметров и снижение влияния деструктивных факторов. В условиях динамического построения ИОТ у преподавателя отсутствует на это время, поэтому такие задачи могут быть делегированы системам искусственного интеллекта.

Анализ электронного обучения в аспектах применения адаптивных подходов и технологий искусственного интеллекта для формирования ИОТ позволяет сделать следующие выводы:

- 1) Искусственный интеллект применяется без учета зависимости прохождения ИОТ от текущего уровня ЗУН и времени.
- 2) Применение технологий искусственного интеллекта в ИЭОС связывается с оптимизацией построения ИОТ, но не описывает взаимного влияния ИИ и деструктивных факторов.

Таким образом применяемые сейчас в ИЭОС технологии ИИ не в полной мере обеспечивают освоения компетенций, предусмотренных учебной программой курса при электронном обучении программированию в установленные сроки с соблюдением предусмотренных образовательным стандартом требований по результатам формирования индивидуальных траекторий освоения обучающимися рабочей программы с учетом недетерминированности характеристик освоения программы обучающимся, обусловленной воздействием деструктивных факторов.

Деструктивные факторы в образовании – это системные проблемы, которые снижают качество обучения, ограничивают доступность знаний и препятствуют развитию обучающихся или усиливают неравенство. Среди деструктивных факторов также необходимо упомянуть устаревшие методы и программы обучения, и нехватку междисциплинарности.

При построении ИОТ в ИЭОС деструктивные факторы необходимо рассматривать как такие внешние и внутренние воздействия на учебный процесс, которые повышают вероятность недостижения требуемого уровня ЗУН в заданные сроки.

Известен методический аппарат учета влияния деструктивных факторов, основанный на работах Д.С. Гнедых [11], С.С. Ермакова [61] и И.Х. Бикмухаметова [89] и обобщённый в работах Дж. Хэтти [104; 105]. Существующие методы снижения воздействия деструктивных факторов требуют больших временных затрат и не могут применяться при динамическом формировании ИОТ без увеличения времени, затрачиваемого преподавателем. В большинстве случаев у преподавателя отсутствует время, которое он может дополнительно потратить на приведение ИОТ конкретного обучающегося в состояние, предусмотренное учебным планом.

Это обуславливает актуальность выявления таких деструктивных факторов, влияние которых может быть снижено путем применения искусственного интеллекта, берущего на себя контактную работу, которую в силу отсутствия времени не может взять преподаватель. Для выявления таких

факторов можно обратиться к работам Дж. Хэтти, в которых выделено около 130 факторов по величине эффекта Коэна  $d$  – статистическому показателю, который измеряет разницу между группами. Некоторые из проанализированных факторов с высоким эффектом Коэна приведены в таблице 1.

Недостатки такого подхода, такие как невозможность точного измерения самооценки знаний обучающихся, отмечены в некоторых работах [92; 129]. Также выражаются сомнения о переносимости результатов анализа на российскую систему образования [54; 74]. Однако при всех недостатках сам перечень факторов в литературе не оспаривается.

Необходимо заметить, что в своей работе Дж. Хэтти рассматривает факторы, влияющие на результаты положительно, поэтому в рамках настоящей работы, для того чтобы рассматривать указанные факторы как деструктивные, проведем логическую инверсию: в колонке «Инверсия описания» приведено описание фактора в деструктивном виде. Например, для фактора  $\varphi_{15}$  «Повышение квалификации» инверсией описания будет «Преподаватель не повышает квалификацию» то есть повышение преподавателем своей квалификации будет положительным фактором, а ситуация, когда преподаватель не повышает квалификацию деструктивным фактором.

Поэтому приведенные в таблице 1 факторы можно рассматривать как основные деструктивные факторы, влияющие на прохождение ИОТ с заданным уровнем ЗУН за предусмотренное время. Для того, чтобы определить зависимость прохождения ИОТ от этих факторов необходимо сгруппировать их и связать с понятиями сферы электронного обучения.

Таблица 1 – Основные факторы, влияющие на результат обучения

№	Фактор	Домен	Обозначение	Значение d	Описание по Хэтти	Инверсия описания
1	2	3	4	5	6	7
1	Самооценка знаний обучающимся	Обучающийся	$\varphi_1$	1,44	Умение учеников критически оценивать свой прогресс и ошибки	Ученики не оценивают свой прогресс и ошибки
2	Программа Пиаже	Обучающийся	$\varphi_2$	1,28	Применение программы Пиаже	Не применяется программа Пиаже
3	Формирующее оценивание	Учебный процесс	$\varphi_3$	0,90	Регулярное проведение контроля знаний и предоставление их результатов	Регулярный контроль и предоставление результатов не проводятся
4	Микрообучение	Преподаватель	$\varphi_4$	0,88	Совместное выполнение заданий вместе с небольшой группой учащихся и их последующий разбор	Совместное выполнение заданий с небольшой группой учащихся и их последующий разбор не проводятся
5	Ускорение в обучении	Учебное заведение	$\varphi_5$	0,88	Опережающее освоение учебной программы способными учениками	Способные ученики не осваивают учебную программу в опережающем режиме
6	Поведение на занятиях	Учебное заведение	$\varphi_6$	0,80	Дисциплина, поведение и эмоциональное состояние участников учебного процесса	Дисциплина, поведение и эмоциональное состояние участников учебного процесса не контролируется
7	Совместное обучение	Учебный процесс	$\varphi_7$	0,77	Совместное обучение учеников с обычными и особыми возможностями	Совместное обучение учеников с обычными и особыми возможностями не осуществляется
8	Ясность	Преподаватель	$\varphi_8$	0,75	Ясность изложения материала преподавателем	Материал не понятен обучающимся
9	Взаимообучение	Учебный процесс	$\varphi_9$	0,74	Возможность обучающимся задавать вопросы, получаться разъяснения, обобщать и в целом вести диалог с преподавателем	Обучающиеся не имеют возможности задавать вопросы, получаться разъяснения, обобщать и в целом вести диалог с преподавателем

Продолжение таблицы 1

1	2	3	4	5	6	7
10	Обратная связь	Учебный процесс	$\varphi_{10}$	0,73	Конструктивные комментарии преподавателя, направленные на улучшение	Преподаватель не дает конструктивные комментарии преподавателя, направленные на улучшение
11	Отношения «учитель-ученик»	Преподаватель	$\varphi_{11}$	0,72	Отношения преподавателя к обучающемуся	Отношения «учитель-ученик» не складываются
12	Индивидуальное или групповое обучение	Учебный процесс	$\varphi_{12}$	0,71	Баланс индивидуальной и групповой работы, в том числе для получения обратной связи	Индивидуальная и групповая работа не ведутся или не сбалансираны
13	Предыдущие достижения	Обучающийся	$\varphi_{13}$	0,67	Предыдущий опыт и достижения обучающегося	Предыдущий опыт и достижения обучающегося не учитываются
14	Самовербализация	Учебный процесс	$\varphi_{14}$	0,64	Самоанализ обучающегося	Самоанализ не проводится
15	Повышение квалификации	Преподаватель	$\varphi_{15}$	0,62	Повышение квалификации преподавателем	Преподаватель не повышает квалификацию

Источник: составлено автором по материалам [104; 105].

Анализ факторов, влияющих на процесс обучения, согласно работе Дж. Хэтти позволяет сделать следующие выводы:

1) в работе приведен объективный список факторов, который путем логической инверсии позволяет получить список деструктивных факторов, однако факторы классифицированы только по доменам и систематически не связаны с понятиями предметной области обучения;

2) зависимость результатов обучения от факторов приведена на основании статистических методов по величине эффекта Коэна, что не позволяет задать аналитическую модель, описывающую зависимость достижения заданного уровня освоения модулей учебных дисциплин в установленные сроки с соблюдением предусмотренных образовательным стандартом требований.

**Модели и функции для описания зависимости достижения заданного уровня освоения модулей учебных дисциплин в установленные сроки с соблюдением предусмотренных образовательным стандартом требований.** В образовании для оценки достижения заданного уровня ЗУН используются различные математические модели и функции, среди них можно выделить следующие:

- логистическая функция моделирует постепенное освоение навыков с ускорением на начальном этапе и замедлением при приближении к максимальному уровню;
- модель Раша используется в тестировании для оценки уровня ЗУН на основе вероятности правильного ответа;
- экспоненциальная модель роста описывает быстрое освоение навыков на начальном этапе с последующим замедлением;
- теория Item Response Theory оценивает вероятность правильного ответа на задание в зависимости от уровня способностей учащегося;
- иерархические линейные модели учитывают влияние факторов разного уровня (индивидуальные, групповые) на прогресс.

Анализ данных моделей и функций показывает, что в свете описанных понятий и их взаимосвязи с выявленными деструктивными факторами наиболее подходящей моделью для аналитической модели, описывающей зависимость достижения заданного уровня освоения модулей учебных дисциплин в установленные сроки с соблюдением предусмотренных образовательным стандартом требований является иерархическая линейная модель. В многоуровневых моделях данные организованы в виде «уровней», на каждом из которых присутствуют общие для всех уровней фиксированные эффекты и случайные эффекты.

Примером иерархической линейной модели может служить двухуровневая модель для оценки достижений учеников. Первый (индивидуальный) уровень описывается по формуле (1)

$$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij}, \quad (1)$$

где  $Y_{ij}$  – результат ученика  $i$  в группе  $j$  (например, оценка);

$\beta_{0j}$  – средний результат группы  $j$ ;

$\beta_{1j}$  – влияние независимой переменной  $X_{ij}$  на результат в группе  $j$ ;

$X_{ij}$  – независимая переменная уровня ученика (например, время подготовки);

$e_{ij}$  – ошибка уровня ученика.

Второй (групповой) уровень описывается по формулам (2) и (3)

$$\beta_{0j} = \gamma_{00} + \gamma_{01}W_j + u_{0j}, \quad (2)$$

$$\beta_{1j} = \gamma_{10} + \gamma_{11}W_j + u_{1j}, \quad (3)$$

где  $W_j$  – независимая переменная уровня группы (например, квалификация учителя),

$\gamma_{00}, \gamma_{01}, \gamma_{10}, \gamma_{11}$  – фиксированные эффекты,

$u_{0j}, u_{1j}$  – случайные эффекты (вариация между группами).

Применение иерархической линейной модели широко рассмотрено в работах таких исследователей, как С. Рауденбуш и А. Брик [123], Л. Хоффман и Р. Ставски [106], Дж. Сингер и Дж. Уиллетт [127], Р. Лент [113], К. Коедингер [111], С. Walkington и М. Bernacki [130], R. Kizilcec [110], X. Cheng [96]. Анализ этих исследований позволяет сделать следующие выводы:

- иерархическая линейная модели широко используется для описания учебного процесса, однако ранее в них не учитывались деструктивные факторы;
- в рассмотренных работах иерархические модели не применялись для задания аналитической зависимости, описывающей прохождение ИОТ.

### **Обзор современных моделей и методов формирования ИОТ.**

Исследователями предложены различные модели и методы формирования индивидуальной образовательной траектории. Среди них можно выделить ряд иерархических моделей, которые, как было показано в параграфе 1.1, используются для описания зависимости достижения заданного уровня освоения модулей учебных дисциплин в установленные сроки с соблюдением предусмотренных образовательным стандартом требований. Например, в работе С.А. Багрецова и Э.В. Мищенко [52] предлагается двухуровневая иерархическая система подготовки специалистов, где первый уровень управляет компетенциями отдельных подсистем (локальных областей), а второй уровень управляет глобальным распределением ресурсов для достижения общего оптимума компетенций. В работе С.А. Багрецова и Т.Э. Черной [51] представлена модель трансформационного процесса индивидуально-группового обучения в вузе, которая основана на поэтапной смене стратегий формирования знаний, умений и навыков. В работе С.А. Багрецова, В.М. Львова, Ю.М. Органова и В.В. Попова [78] введена двухуровневая иерархическая модель управления развитием компетенций,

которая позволяет планировать подготовку и доподготовку персонала с учётом временных и стоимостных затрат.

Оценка знаний, умений и навыков является одним из основных параметров, влияющих на построение ИОТ. В исследованиях описаны как методы оценки профиля обучающегося [48; 69], так и методы динамического оценивания текущих или промежуточных результатов обучения [49; 50; 64].

При динамической оценке ЗУН задача построения ИОТ может решаться как задача нахождения пути на графе. Например, в работе [70] рассматривается методика выбора оптимальных вариантов подготовки специалистов на этапе ввода в строй с использованием методов векторной оптимизации. Подготовка специалистов представляется в виде ориентированного графа, где вершины соответствуют темам обучения, а дуги описывают возможные связи между ними с учётом временных, финансовых и ресурсных затрат. Оптимизация траектории сводится к задаче поиска кратчайшего пути с учётом множественных критериев, решаемой методом скаляризации и булевого программирования для исключения циклов.

Однако, оптимизация траектории может требовать применения сложных стратегий, для которых могут использоваться марковские процессы принятия решений [25; 66; 67; 68; 76; 87].

Необходимо заметить, что вторым важнейшим параметром, учитываемым при построении ИОТ, является время, рациональное распределение которого, существенно влияет на достижение заданного уровня ЗУН. Так в работе А.И. Алтухова [46] предложен метод априорного оценивания качества использования учебно-тренажёрных комплексов при проведении практических занятий, который позволяет определить рациональное распределение времени на этапах занятий, выявить ошибки и выработать рекомендации преподавателю по улучшению методики.

Современные подходы к подготовке и оценке специалистов характеризуются высокой степенью методологического разнообразия и

направлены на повышение индивидуализации образовательных траекторий. Интерес представляют иерархические модели управления компетенциями, позволяющие учитывать локальные и глобальные показатели развития. Применение полумарковских процессов [49] демонстрирует эффективность стохастического моделирования для анализа переходов между этапами обучения и оптимизации временных и ресурсных затрат. Работы, посвящённые агрегированию оценок знаний и построению адаптивных траекторий подготовки [26; 27; 28; 43], подчеркивают важность диагностики индивидуальных особенностей обучающихся и гибкой настройки содержания обучения. Использование методов многокритериальной оптимизации, нечёткой логики и дискриминантного анализа способствует более точной оценке компетенций и повышению качества подготовки специалистов.

## **1.2 Научно-методический аппарат организации управления процессом электронного обучения на основе индивидуализации и адаптации**

**Модели адаптивного обучения.** Как было сказано выше, адаптивное обучение является разновидностью обучения, при котором для обучающегося формируется индивидуальная траектория обучения, опирающаяся на непрерывный анализ его персональных учебно-значимых характеристик с целью обеспечить эффективные, действенные и индивидуальные траектории обучения для вовлечения каждого обучающегося». Технологии, основанные на принципах адаптивного обучения, разработаны для обеспечения эффективных, действенных и индивидуальных траекторий обучения для вовлечения каждого обучающегося.

В адаптивном обучении широко применяется искусственный интеллект. Л. Пульезе [120] приводит основные признаки ИЭОС и на основании этого выделяет следующие типы ИЭОС адаптивного обучения:

- адаптивные системы на основе машинного обучения;

- адаптивные системы с расширенными алгоритмами;
- адаптивные системы на основе правил;
- адаптивные системы на основе дерева решений.

Российские и зарубежные исследователи предлагают различные подходы к реализации указанных типов ИЭОС. В первую очередь предлагаются различные варианты адаптивности. Так А.Н. Полетайкин, С.Г. Синица, Е.Ю. Кунц описывают технологию организации системы модульного адаптивного обучения с верификацией компетенций на основе онтологий [75].

Предлагаются и другие подходы. Например, в диссертациях Д.С. Ботова [18] и В.Н. Нгуен [14] предложены графовые модели образовательного процесса. В дальнейшем к этим моделям применяются методы имитационного моделирования и нейросетевые модели.

Некоторые авторы, как И.Е. Николаева [16], М.С. Гаспариан, С.А. Лебедев, Ю.Ф. Тельнов [56] основывают свои модели на требованиях рынка труда на уровне знаний и компетенций и предлагают алгоритмы формирования образовательных программ по профилям подготовки на основе анализа обобщенных трудовых функций и трудовых функций профессиональных стандартов.

Широко представлены компетентностные модели. О. В. Махныткина в диссертационной работе [21] предложила модель оценки компетентности студента вуза с учетом требований и ограничений ФГОС ВПО третьего поколения, а также модель оптимизации ИОТ на основе комплексного использования гибридного подхода, методов предобработки данных и динамического программирования. Т.А. Ткалич и Е.Г. Гриневич рассматривают подходы к формированию адаптивных образовательных программ, систематизирующих и структурирующих компетенции современного ИТ-специалиста и представляют основанную на европейских

моделях компетенций EQF и e-CF схему верхнего уровня онтологии образовательного процесса.

Однако в данных работах в основном рассматривается вопрос адаптации ИУП на основе профиля обучающегося и различных компетентностных и онтологических моделей без динамической корректировки ИОТ, а предложенные модели и методы не позволяют задать аналитические зависимости для динамической корректировки ИОТ с учетом деструктивных факторов.

**Существующие методы построения индивидуальной образовательной траектории.** Анализ опыта применения ИОТ показал, что каждая индивидуальная траектория должна формироваться динамически на основе оценки результатов взаимодействия педагога и обучающегося на индивидуальном и групповом уровне линейной иерархической модели с учетом выявленных групп деструктивных факторов.

Вопросы формирования ИОТ рассматривался исследователями с точки зрения разных подходов.

А.А. Власенко [19] в диссертационной работе предложены структура и алгоритмическое обеспечение адаптивной системы дистанционного обучения в сфере информационных технологий с использованием компетентностного подхода, позволяющие на основе модели обучающегося осуществлять динамическое изменение индивидуального учебного плана на любом этапе обучения, а также предложен двухэтапный алгоритм формирования учебного плана, включающего совокупность дисциплин, покрывающих заданный набор компетенций, отличительной особенностью которого является эвристическая процедура выбора дисциплин, основанная на жадной стратегии.

В работе М.А. Косоноговой [12] разработан метод управления индивидуальной траекторией обучения, который позволяет производить адаптивное формирование блока объектов учебного контента на каждом шаге обучения, учитывая уровень подготовленности обучающегося, его

потенциальные возможности, определяемые зоной ближайшего развития, и социотип.

И.А. Кречетов [13] предложил модель адаптивного обучения, агрегирующая модель забывания, модель предметной области, модель обучаемого, и позволяющая автоматизировать процесс адаптивного обучения, получен алгоритм определения индивидуальной траектории обучения в системе дистанционного обучения, а также разработана система организации процесса адаптивного обучения, автоматически формирующая траектории, адаптируя их на основе изменения состояний модели обучаемого.

Е.Ю. Благов, И.А. Лещева, С.А. Щербан описывают алгоритм формирования «дополнительных» компетенций, на основе которого ими предложена усовершенствованная онтологическая модель компетентностно-ориентированного учебного плана, включающая в себя внешние источники «дополнительных» компетенций для своевременной коррекции содержания учебных планов в соответствии с изменением запросов актуальных и потенциальных работодателей соответствующих отраслей экономики [53].

В работе О.Х.З. Аль-Дулаими [10] предложена структурная модель корпоративной организационной системы подготовки обучаемых, обеспечивающая заданный уровень качества результатов обучения на основе оптимального варианта мотивации и планирования за счет достижения равновесия в доминантных стратегиях агентов по их затратам для достижения установленных компетенций. На основе данной модели разработан программный комплекс управления качеством подготовки обучаемых в рамках корпоративных организационных образовательных систем, обеспечивающий заданный уровень форсированности компетенций на базе нейросетевых моделей представления знаний.

Р.Е. Медведевым [22] разработана онтологическая модель знаний, описывающая знания о предметных областях ЭОС, создана семантическая

метрика для контроля процесса обучения на основе формального анализа контрольных точек ИОТ на основе чего показана взаимосвязь ИОТ с онтологической системой представления знаний о предметной области, дающая возможность разделить процессы формирования кластеров учебного материала и ИОТ.

В работе Ф.Х. Нгуен [15] предложен способ унифицированного представления информации о критериях, позволяющих оценить качество реализации образовательной программы электронного обучения, собранной из двух независимых источников а также разработана схема управления программами электронного обучения с обратной связью, предполагающая управление по четырем каналам на основе информации, учитывающей мнение респондентов о качестве реализации программ обучения в ЭОС и применении управляющих воздействий для повышения их качества.

В работе А.А. Сеньковской [24] предложен математический метод моделирования учебного процесса с целью анализа и рационализации учебных планов.

В работе А.В. Макарова [20] предложена формальная информационная модель интегративного семантико-логического представления содержания обучения и результатов образовательной деятельности учащихся. Модель основана на синтезе семантических сетей и объектно-ориентированного подхода, а также применении словаря смысловых отношений и использовании механизма структурных паттернов. Также разработан метод и алгоритм расчета уровня семантико-логической связности результатов диагностики предметной подготовленности обучаемого, основанный на объединении и дальнейшем семантико-логическом анализе структуры контрольно-измерительного материала и результатов диагностики предметной подготовленности обучаемого. Метод направлен на выделение и последующую обработку смысловых цепочек контрольных заданий.

В работе О.В. Перченок [23] разработана математическая модель самопроверяемых задач, которая приспосабливает структуру условия задачи для автоматической верификации и устанавливает взаимосвязи между математическими объектами «самопроверяемая задача», «абстрактный класс самопроверяемых задач», «предметно-ориентированный класс самопроверяемых задач», «верификатор задачи», также описан алгоритм автоматической верификации решений.

Указанные авторы под процессом принятия решений о корректировке ИОТ и, соответственно, под формированием ИОТ понимали разработку учебных планов, соответствующих формализованным требованиям (стандартам, статистике, профилю обучающегося) на момент составления плана с последующей оценкой прохождения обучающимся траектории. Такой подход не в полной мере учитывает недетерминированность деятельности обучающегося по освоению предмета, что ведет к отклонению пройденной ИОТ от ИОТ, заданной при составлении плана. Возникновение таких отклонений деструктивно влияет на качество и сроки освоения предмета.

Учет недетерминированности [115] процесса обучения позволяет применить так называемый [2] агентный подход, в котором участники образовательного процесса (преподаватель, студент и другие.) описываются как интеллектуальные агенты, которые выбирают действия, максимизирующие меру своей эффективности, основываясь на воспринимаемой информации и знаниях, хранящихся в памяти.

При такой formalизации участники процесса обучения – это множество агентов, образующих многоагентную систему, в которой их взаимодействие регулируется протоколами коммуникации и правилами принятия решений и направлено на достижение общей цели [68; 72], под которой понимается разработка эффективного метода управления индивидуальной образовательной траекторией в виде комплекса проблемно-ориентированных программ, обеспечивающих достижение

требуемого уровня знаний, умений и навыков за установленное время при обучении программированию в интеллектуальной электронной образовательной системе.

Однако для того, чтобы такая многоагентная система эффективно функционировала в условиях недетерминированности процесса обучения, она должна обеспечивать оперативную корректировку ИОТ в зависимости от поступающей в режиме реального времени информации о текущих результатах обучения, что обуславливает необходимость разработки комплексной информационной модели, позволяющей решить в ИЭОС следующие задачи:

- оперативно скорректировать ИОТ на основе оценки текущего состояния достижения требуемого уровня профессиональных компетенций;
- динамически изменить ИОТ в зависимости от изменения сценария учебного процесса;
- провести планирование действий участников образовательного процесса на основе динамически измененной ИОТ.

**Применение агентного подхода при построении индивидуальной образовательной траектории.** Как было сказано выше, оперативная корректировка ИОТ может быть обеспечена использованием агентного подхода, при котором участники образовательного процесса взаимодействуют в рамках многоагентной системы.

Существуют различные подходы к типизации многоагентных систем. В работе C.A. Iglesias, M. Garijo, J. C. Gonzalez [114] предлагалась следующая классификация:

- 1) объектно-ориентированные методологии, в которых агенты рассматриваются как активные объекты;
- 2) методологии инженерии знаний, в которых используются онтологии и решатели задач;

3) формальные подходы, основанные на применении формальных языков и темпоральной модальной логики;

4) смешанные варианты.

Данная классификация получила развитие в работе В.Б. Тарасова [82], который предложил более подробный вариант классификации:

1) моделирование индивидуального и коллективного поведения агентов в многоагентной системе;

2) теоретико-игровое моделирование взаимодействия агентов;

3) лингвистическое направление, связанное с моделированием речевых актов для построения протоколов коммуникации между агентами;

4) организационное моделирование многоагентных систем, связанное с общей теорией систем и организаций;

5) построение моделей распределенных систем на базе сетей Петри;

6) программистское направление, связанное с моделями акторов, параллельным и агентно-ориентированным программированием;

7) логико-семиотическое направление, связанное с моделированием ментальных свойств агентов с помощью неклассических логик.

А.В. Кузнецов обобщает [63] приведенные классификации, предлагая следующее условное деление:

1) системы агентов с непрерывным и дискретным временем, которое может быть как общее для всех агентов системы, так и разное для разных агентов;

2) системы агентов на сетке и в обычном евклидовом пространстве и, возможно, в других топологических пространствах;

3) системы, в которых агенты перемещаются по некоторому пространству и системы, в которых пространство состоит исключительно из агентов;

4) системы, в которых агенты являются исключительно виртуальными сущностями среды моделирования и в которых агенты являются реальными устройствами, например, радиостанциями.

Ю.Ф. Тельнов и В.А. Казаков в своих исследованиях разрабатывали модели интеграции образовательных ресурсов через агентные технологии [79]. Ими был создан Прототип многоагентной системы сетевого взаимодействия учебных заведений с применением фреймворка JADE и стандартов FIPA, эта система позволяет динамически формировать образовательные программы, учитывая запросы рынка труда и индивидуальные потребности студентов. Каждый агент в системе отвечает за конкретные задачи: анализ компетенций, подбор курсов, интеграцию ресурсов из разных вузов. Архитектура системы включает онтологию информационно-образовательного пространства, преобразованную из формата Web Ontology Language для взаимодействия агентов.

В работе Т.М. Шамсутдиновой [90] описана система на основе модели студента и адаптивных алгоритмов, в которой используют агенты для анализа данных о студенте (стиль обучения, успеваемость, предпочтения) и адаптации контента.

При сравнении различных платформ адаптивного обучения [27] может быть выделена разработанная Томским государственным университетом система Plario, которая использует алгоритмы ИИ для адаптации курсов математики. Система анализирует ошибки студентов и автоматически предлагает дополнительные материалы, имитируя работу тьютора.

В контексте настоящей работы важно отметить, что классические многоагентные системы в основном построены на коммуникации агентов с использованием формальных языков, например FIPA, KQML или ACL. Лингвистическое направление во взаимодействии агентов выделяется исследователями, но до последнего времени не находило широкого распространения.

Анализ приведенных работ показывает, что использование формальных протоколов для коммуникации агентов требует формализации описания программ и содержания учебных курсов, что является затруднительным в силу их объема и изменчивости. В связи с этим актуализируется вопрос применения лингвистического направления взаимодействия агентов. При лингвистическом взаимодействии естественным способом выражения агента становится использование больших языковых моделей.

**Применение больших языковых моделей при построении индивидуальной образовательной траектории.** Исследования применения такого вида нейронных сетей глубокого обучения как большие языковые модели при построении ИОТ могут быть разделены на две группы:

- Исследования применения больших языковых моделей;
- Исследования комбинированного подхода, сочетающего большие языковые модели и обучение с подкреплением.

К первой группе могут быть отнесены работы С. Chen, A. Leitch [93], в которой исследуется потенциал больших языковых моделей в качестве академических помощников для чтения. Авторы описывают использование интеллектуального агента на основе Anthropic's Claude.ai и проводят сравнительный анализ контрольной и экспериментальной групп, использующих БЯМ в учебном процессе, оценивают практическую пользу такой интеграции, а также работу J. Lu и соавторов [116], которые предложили систему YODA, реализующую прогрессивный цикл обучения через взаимодействие агента-учителя (предоставляет обратную связь) и агента-ученика (улучшает ответы), что позволяет эффективно усваивать материал на основе ограниченных примеров.

Ко второй группе исследований могут быть отнесены работы следующих исследователей. S. Xu, X. Zhang, L. Qin представили [132] фреймворк EduAgent, который использует БЯМ GPT-3.5 и GPT-4 для создания

агентов, моделирующих поведение студентов. Важным аспектом является интеграция когнитивных критериев, позволяющих БЯМ выявлять корреляции между поведением и успеваемостью, что можно рассматривать как шаг к более глубокому, «осознанному» моделированию. B. Radmehr, A. Singla, T. Käser в своей статье [122] расширяют возможности обобщения агентов в открытых текстовых обучающих средах за счет интеграции обучения с подкреплением с НГО. Авторы рассмотрели два метода интеграции помощи НГО: Suggestion-Assisted RL и Decision-Assisted RL. В методе SA-RL, НГО предлагает несколько действий, из которых RL-агент выбирает, тогда как в DA-RL, НГО выбирает лучшее действие из предложенных RL-агентом.

Анализ приведенных выше исследований показывает, что их центральным элементом была оценка качества и достоверности данных, которые были сгенерированы БЯМ. Этот вопрос актуален, поскольку проблема так называемых галлюцинаций БЯМ общеизвестна и на текущий момент не имеет полного решения.

Однако, необходимо заметить, что в сфере точных наук указанные проблемы имеют решение. Поскольку для большинства точных наук при решении задачи может быть задан формальный критерий проверки правильности. Пока такой критерий не будет выполнен, то задача не считается решенной.

Примером подхода с формальным критерием правильности является предложенный исследователи из Google DeepMind алгоритм FunSearch который с помощью модели Codey генерирует Python-программу, реализующую с учетом некоторых ограничений ранее не предлагавшееся решение так называемой The Cap Set Problem. Модели на вход подается простое решение и описание проблемы, модель генерирует несколько разных решений, которые валидируются. Решения, прошедшие проверку, снова подаются на вход модели с запросом на формирование улучшенного решения, итерация повторяется. Подход имеет ограничения: должна быть доступна

легковесная числовая функция валидации решения, а решение должно быть представлено в виде кода или текста.

Также в качестве примера системы с возможностью формальной проверки можно привести разработанную С. Zhang и соавторами [133] систему Mathemyths.

Таким образом можно сделать вывод о возможности применения в некоторых областях больших языковых моделей в многоагентных системах, формирующих ИОТ. Одной из таких областей является обучение программированию. Также необходимо отметить, что среди приведенных примеров многоагентных ЭОС, использующих БЯМ, отсутствовали системы обучения программированию.

### **1.3 Анализ исследований в области адаптивного обучения программированию**

**Применение больших языковых моделей в адаптивном обучении программированию.** J. Chen и соавторы представляют [95] чат-бота NetLogo, предназначенного для изучения и практики в языке NetLogo, предназначенном для агентно-ориентированного моделирования. Основываясь на опыте авторов в обучении NetLogo, они создали прототип, основанный на конструктивистской теории обучения, с акцентом на поддержку пользователей в итеративном создании своих подсказок и небольших фрагментов кода перед началом работы над целыми моделями.

C. Lee и J. Xiong в своей работе [112] предлагают методологию с интеграционной платформой на базе искусственного интеллекта для эффективного обучения детей основам компьютерного мышления. Система, названная Spark, использует гибридную педагогику, которая поддерживает как нисходящий, так и восходящий подходы к обучению. Дети могут описать желаемую задачу на естественном языке, а система ответит простой для понимания программой, состоящей из подзадач нужного уровня сложности.

Робот может немедленно выполнять программу, демонстрируя результаты детям.

A. Salmon, K. Hammer и соавторы исследуют [125] как различные стратегии запросов к большим языковым моделям влияют на эффективность объяснения ошибок программирования. Авторы рассматривают возможность использования БЯМ для генерации объяснений ошибок без предоставления исходного сообщения об ошибке, которое обычно выдается компилятором или интерпретатором.

В работе P. Denny, V. Kumar и N. Giacaman [98] исследуют, как GitHub Copilot, основанный на модели OpenAI Codex, справляется с задачами уровня CS1. Они проверили GitHub Copilot на 166 задачах на языке Python, касающихся работы с условными операторами, строками, списками и двумерными массивами. Для этого подавали на вход БЯМ описание проблемы, проверяли и уточняли задачу, если сгенерированный код не проходил тесты. Исследование показало, что Copilot смог решить 79 задач (47,6%) с первой попытки. После уточнения описания задач, Copilot успешно справился еще с 53 из оставшихся 87 задач (60,9%).

В другом исследовании P. Denny и соавторы [99] исследуют влияние генеративного ИИ на образование в области компьютерных наук. Они подчеркивают, что хотя генеративный ИИ предлагает значительные возможности для обучения, существует опасение, что студенты могут чрезмерно полагаться на эти инструменты, не участвуя в процессе обучения.

В исследовании, проведенном C. Kerslake, J. Prather, P. Denny и соавторами [107], рассматривается интеграция задач, связанных с созданием запросов на естественном языке, в курсы введения в программирование. Авторы предлагают два типа задач: «Explain in Plain English», где студенты описывают функции на естественном языке, и «Prompt Problems», где студенты создают запросы для генерации кода, эквивалентного заданному фрагменту.

В своей работе Н. Keuning, A. Luxton-Reilly и их соавторы [108] рассматривают потенциал генеративного ИИ в преподавании вводного курса программирования CS1. Авторы отмечают, что генеративный ИИ уже способен генерировать правильный код для начального уровня задач, и его возможности продолжают улучшаться. Они предлагают использовать эти способности для повышения мотивации студентов, расширения их понимания процесса разработки программного обеспечения и вовлечения в более аутентичное обучение.

В рамках исследования, проведенного более чем на 100 студентах, N. Dunder и соавторы [101] отобрали из инструмента автоматической генерации кода и его оценки системой «Kattis» 127 случайных задач на Python, распределенных по трем уровням сложности. Каждое задание по кодированию от системы «Kattis» отправлялось в ChatGPT-3.5. Из 127 решений только 19 (15%) были полностью приняты «Kattis», из них 10 решений относились к первому уровню сложности. Для 108 задач с неправильным по оценке Kattis решением 83 (77%) были классифицированы как «неправильный ответ», 16 (15%) как «ошибка времени выполнения» и 9 (8%) как «превышен лимит времени». Из 108 неправильных решений 19 (15% от общего числа) были классифицированы как частично принятые, то есть прошедшие проверку на определенной части тестовых данных. Была обнаружена положительная корреляция (0,43) между одобренными решениями и количеством попыток и отрицательная корреляция (-0,48) между уровнем сложности и количеством одобренных решений.

N. Kiesler и D. Schiffner [109] провели исследование производительности ChatGPT-3.5 и GPT-4 при решении задач вводного курса программирования на Python. Для этого из открытой платформы CodingBat были взяты 72 задачи. БЯМ продемонстрировали от 94,4% до 95,8% правильных решений. Однако авторы подчеркивают, что типовые решения для всех задач CodingBat доступны на GitHub, что позволяет предположить,

что вероятность того, что ChatGPT был обучен на таких данных, высока. Исследование также показало, что обе БЯМ предоставляли текстовые объяснения и программный код в 97,2% случаев, что может быть полезно для обучения программированию.

J. Savelka и соавторы [126] оценили способность БЯМ DaVinci (text-davinci-002 и text-davinci-003) на базе ChatGPT сдавать экзамены на вводных и промежуточных курсах программирования на Python по таким темам как «введение в программирование», «типы данных», «переменные», «ввод-вывод, базовые операции», «условия, циклы, логические операции», «функции и структуры данных», «модули и пакеты», «строки», «ООП». Результаты показали, что из 530 задач text-davinci-002 правильно решил 301 (56,8%), а text-davinci-003 – 341 (64,3%). Хотя прогресс в версиях БЯМ очевиден, но проходной балл для данных тестов составляет 70%, и авторы делают вывод о том, что DaVinci не прошел экзамен.

Анализ работ позволяет сделать вывод о том, что адаптивное обучение программированию при агентном подходе в основном сосредоточено на использовании одного агента. Это не позволяет в полной мере смоделировать учебный процесс и решить задачи оперативной корректировки ИОТ и динамической адаптации.

**Обзор существующих информационных систем адаптивного обучения программированию с применением больших языковых моделей.** В литературе описаны различные подходы к созданию информационных систем адаптивного обучения программированию с применением больших языковых моделей.

Широко применяемая ИЭОС Moodle имеет средства построения ИОТ в зависимости от результатов пройденных тестов [43]. Также в Moodle реализована подсистема искусственного интеллекта, которая позволяет интегрироваться с различными нейронными сетями глубокого обучения и может быть использована при решении таких задач, как генерация тестовых

заданий или реализация диалогового агента, помогающего в объяснении материалов курса [44; 45].

C. Qian и соавторы описывают [121] парадигму многоагентного обучения, разработанную для расширения возможностей агентов в интерактивном решении задач за счет использования накопленного опыта. В статье описывается фреймворк, который объединяет два типа языковых агентов (инструктор и ассистент) с помощью специализированных модулей совместного отслеживания, запоминания и мышления.

В работе P. Denny и соавторов [102] представляют подход к обучению программированию с применением нейронных сетей глубокого обучения, основанную на упражнениях, направленных на развитие навыков создания эффективных запросов для генерации кода с помощью больших языковых моделей. Для реализации этого подхода был разработан веб-инструмент Promptly, который позволяет студентам формулировать запросы для БЯМ, на основе подсказок генерирует фрагменты Python-кода и проверяет сгенерированный код в инструменте CodeRunner.

Необходимо отметить работу группы исследователей под руководством T. Wang, которые создали многоагентную систему GenMentor, где каждый агент на базе БЯМ выполняет специализированную задачу:

- Агент идентификации навыков, обученный с применением Chain of Thought на датасете «цель → навыки» сопоставляет цели обучения с необходимыми навыками.
- Агент профилирования обучающегося создает динамический профиль студента, включающий уровни владения навыками, предпочтения и поведение.
- Агент планирования траектории создает персонализированную учебную траекторию, которая адаптируется по мере прогресса и в результате симуляции может предсказать реакцию студента на контент и скорректировать траекторию.

– Агент создания контента генерирует учебные материалы через механизм выборки с дополнением.

Эксперименты показали, что GenMentor на 20–30% точнее базовых методов (например, Direct Prompt) в определении необходимых навыков, а качество траекторий обучения по таким метрикам как «Прогрессия» и «Вовлеченность» выше. Участники тестирования отметили, что GenMentor улучшил их эффективность (85% участников) и повысила персонализацию контента (80% участников).

Однако в описанных выше информационных системах отсутствует формализованное описание критерия достижения целей взаимодействия, что не влияет на качество обучения, но не позволяет дать ответ на вопрос: будет ли достигнут требуемый (иногда минимальный) уровень профессиональных компетенций за отведенное рабочей программой время.

Говоря буквально, ИЭОС, подобные GenMentor, позволят упрочить свои успехи студентам-отличникам, подтянуться студентам-хорошистам, но не дадут преподавателю ответы на такие вопросы как, освоит ли в текущем семестре минимально необходимый уровень студент-троечник и какие действия и за какое время можно предпринять, чтобы хорошист достиг уровня отличника.

Анализ указанных работ позволяет сделать вывод о том, что несмотря на интенсивное развитие систем адаптивного обучения программированию, использующих многоагентный подход, существующие системы не оперируют понятием ИОТ, поэтому не позволяют динамически изменять ИОТ в зависимости от скорректированного сценария и учитывают только незначительную часть деструктивных факторов.

#### **1.4 Постановка задачи исследования**

Целью настоящей работы является разработка метода управления индивидуальной образовательной траекторией в виде комплекса

проблемно-ориентированных программ, обеспечивающих достижение требуемого уровня знаний, умений и навыков за установленное время при обучении программированию в интеллектуальной электронной образовательной системе. При этом должна быть учтена недетерминированность процесса обучения.

Под недетерминированностью процесса обучения в настоящей диссертации понимается невозможность однозначно предсказать на основе полного знания начальных условий и правил будущее состояние системы «Преподаватель-Обучающийся» в связи с существованием альтернатив на каждом этапе формирования ИОТ, влиянием случайных и неслучайных деструктивных факторов и невозможности алгоритмического предсказания некоторых процессов.

Для достижения поставленной цели необходимо решить общую научную задачу: разработать модели и методику, обеспечивающие достижение обучающимся требуемого уровня знаний, умений и навыков в установленные образовательным стандартом сроки, за счет динамической корректировки индивидуальной образовательной траектории с учетом деструктивных факторов.

Для решения общей научной задачи ее необходимо декомпозировать на следующую совокупность взаимоувязанных частных научных задач:

1) разработка модели взаимодействия участников учебного процесса для оперативной корректировки индивидуальной образовательной траектории, учитывающей соблюдение заданного уровня знаний, умений и навыков и сроков обучения и включающей динамически формируемую на основе учебно-методических материалов базу знаний;

2) разработка методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию; и частную прикладную задачу:

3) разработка практических рекомендаций по созданию комплекса проблемно-ориентированных программ для оперативной корректировки индивидуальной образовательной траектории с учетом соблюдения заданного уровня знаний, умений и навыков и сроков обучения.

Взаимодействие участников учебного процесса, направленное на достижение заданного уровня знаний, умений и навыков  $R$  курса в установленные сроки с соблюдением предусмотренных образовательным стандартом требований с учетом недетерминированности характеристик освоения программы обучающимся может быть описано как зависимость от времени  $T$  в виде формулы (4) и неравенств (5)

$$R = f(T), \quad (4)$$

$$R^{min} \leq R \leq R^{max}, T^{min} \leq T \leq T^{max}, \quad (5)$$

где  $R$  – фактическая оценка уровня ЗУН в баллах при электронном обучении программированию;

$R^{min}$  – удовлетворительная оценка уровня ЗУН в баллах согласно образовательной программе курса;

$R^{max}$  – максимальная оценка уровня ЗУН в баллах согласно образовательной программе курса;

$T$  – фактическое время (в часах) изучения курса;

$T^{max}$  – максимально допустимое время изучения курса (в часах), включающее регламентированное время изучения курса согласно образовательной программе дисциплины и время, затраченное преподавателем на работу по курсу;

$T^{min}$  – минимальное время (в часах) изучения курса, может быть оценено как время контактной работы  $t^{Kont}$ .

Задача достижения заданного уровня ЗУН при соблюдении условия  $R^{min} \leq R \leq R^{max}, T^{min} \leq T \leq T^{max}$  может быть сформулирована как нахождение оптимального варианта ИЭОС  $B$ , описываемого функцией (6)

$$\text{optimum}(B), \quad (6)$$

где  $B$  – интеллектуальная электронная образовательная система, описываемая формулой (7)

$$B = \langle A^S, A^L, C \rangle. \quad (7)$$

где  $A^S$  – множество обучающихся;  
 $A^L$  – множество преподавателей;  
 $C$  – множество изучаемых курсов.

Необходимо заметить, что применение ИЭОС  $B$  является обязательным условием, поскольку сбор всех необходимых для достижения цели диссертации параметров в ручном режиме является крайне затруднительным.

Нахождение  $\text{optimum}(B)$  должно обеспечивать выполнение критерия оптимальности (8)

$$(R < R^{\min} \wedge T^{\max} - T \leq \varepsilon) \vee (R^{\min} < R \leq R^{\max} \wedge T^{\max} - T > \varepsilon), \quad (8)$$

где  $\varepsilon$  – порог риска, то есть количество часов, после которого достижение заданного уровня ЗУН за регламентированное время считается невозможным.

При этом должно соблюдаться условие, что за заданное время  $T^{\max}$  должен быть достигнут минимальный уровень ЗУН модулей учебных дисциплин  $R$ , а в случае, если фактическое время  $T$  меньше регламентированного времени  $T^{\max}$  уровень ЗУН  $R$  должен максимизироваться до момента достижения  $T^{\max}$  или момента  $R = R^{\max}$ .

## **Выводы по первой главе**

В первой главе проанализированы результаты современных российских и зарубежных исследований в области электронного и

адаптивного обучения, а также интеллектуальные электронные обучающие системы. Показано, что данные исследования не в полной мере обеспечивают решение актуальной задачи достижения заданного уровня ЗУН за регламентированное время.

Рассмотрены существующие алгоритмы, методы и системы, формирующие ИОТ с применением МППР, многоагентности и нейронных сетей глубокого обучения. Проведен анализ существующих моделей влияния деструктивных факторов на учебный процесс с учетом достижения заданного уровня ЗУН в установленные сроки с соблюдением предусмотренных образовательным стандартом требований.

Анализ существующих ИЭОС и методов формирования и корректировки ИОТ, в том числе в сфере обучения компьютерным наукам, показал, что применение классических моделей и аналитических методов формирования и корректировки ИОТ сталкивается с принципиальными трудностями, обусловленными необходимостью описания сложных зависимостей в условиях постоянного изменения учебно-методических материалов и подходов к учебному процессу.

В ряде существующих научно-методических аппаратов для формирования ИЭОС широко применяются технологии искусственного интеллекта, обладающие определенными возможностями по адаптации существующих образовательных программ к особенностям обещающегося, однако их применение не позволяет оптимально распределить время и учесть влияние деструктивных факторов при динамической корректировке ИОТ для обеспечения достижения заданного уровня ЗУН за регламентированное время.

В проанализированных работах описываются в основном аналитические методы описания зависимостей и многокритериальные задачи принятия решений, однако приведенные там модели не описывают комплексно зависимость принятия решения о выполнении действий от распределения времени и текущего уровня ЗУН в условиях влияния

конкретных деструктивных факторов и недетерминированности действий участников учебного процесса.

Таким образом задача исследования актуальна и связана с комплексным применением нейросетей глубокого обучения в марковском процессе принятия решений для достижения заданного уровня ЗУН за регламентированное время путем динамической корректировки индивидуальной образовательной траектории.

## Глава 2

### Модель взаимодействия участников учебного процесса для оперативного формирования ИОТ

#### **2.1 Обоснование требований к модели взаимодействия участников учебного процесса для оперативного формирования ИОТ в целях достижения заданного уровня ЗУН за регламентированное время**

Как показано в первой главе, достижение заданного уровня ЗУН за регламентированное время может быть описано как процесс взаимодействия преподавателя и обучающегося в условиях воздействия деструктивных факторов и характеризуемый аналитической зависимостью заданного уровня ЗУН от времени.

Анализ предметной области показал, что для описания такой зависимости применяются различные варианты моделей: логистическая функция, модель Раша, экспоненциальная модель, Item Response Theory, иерархические линейные модели.

В настоящей диссертации учебный процесс рассматривается с точки зрения системы «Преподаватель – Обучающийся», в которой можно выделить два уровня: индивидуальный и групповой. Что обуславливает выбор иерархических моделей для более подробного рассмотрения.

Результаты анализа существующего научно-методического аппарата [96; 106; 110; 111; 113; 123; 127; 130], приведенные в главе 1 показали, что иерархические модели являются наиболее подходящими для описания аналитическим способом различных взаимодействий в рамках учебного процесса, например «Студент-Факультет», «Навык аспиранта – Научный руководитель», «Годы – Кафедра», «Прохождение теста – особенности курса», «Навык – Стратегия поддержки». Однако существующие иерархические

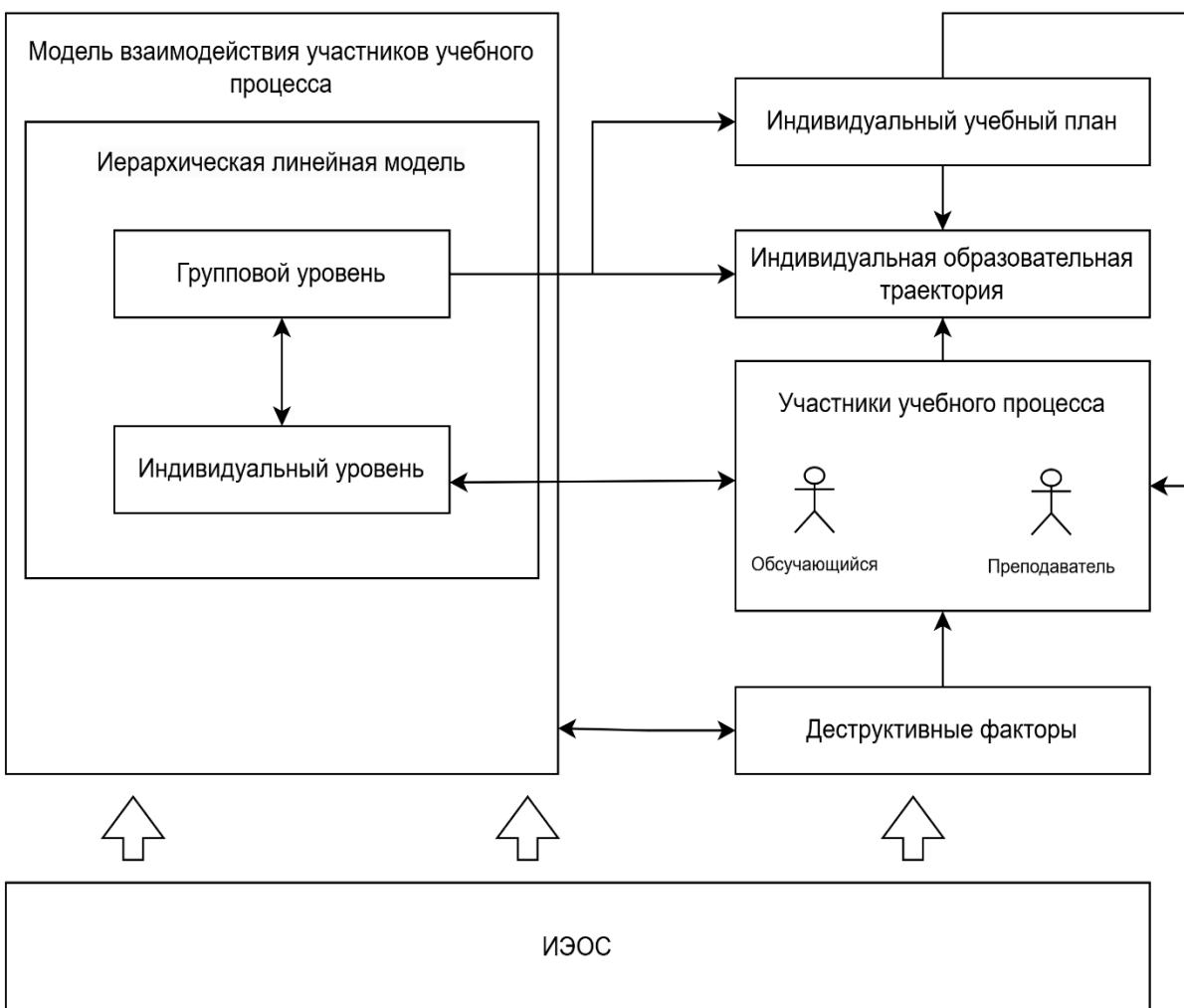
модели описывают зависимости только аналитическим путем, что не позволяет в должной мере учесть влияние деструктивных факторов и описать недетерминированность учебного процесса, под которой в диссертации понимается совокупность следующих особенностей:

- невозможность однозначно предсказать на основе полного знания начальных условий и правил будущее состояние системы «Преподаватель-Обучающийся»;
- существование альтернатив на этапах формирования ИОТ;
- влияние случайных и неслучайных деструктивных факторов;
- невозможность алгоритмического предсказания некоторых процессов.

В этой связи определим, что модель, решающая задачу построения аналитической зависимости для синтеза методики формирования ИОТ, должна обеспечивать проведение исследований в части:

- использования нейронных сетей глубокого обучения для описания зависимостей в рамках предложенной Л. Заде концепции «мягких вычислений», заключающейся в применении вероятностных и эвристических методов, терпимости к неполноте и неточности данных для получения решений в сложных нечетких условиях высокой неопределенности [134];
- формирования двух уровней иерархии: индивидуального, включающего преподавателя и обучающегося по отдельности и группового, на котором они взаимодействуют.

Схема такой модели приведена на рисунке 4. Схема описывает взаимодействие участников учебного процесса на двух уровнях, а пространство состояний ИЭОС задается ИУП, а результат прохождения каждого этапа в условиях воздействия деструктивных факторов формирует ИОТ.



Источник: составлено автором.

Рисунок 4 – Схема двухуровневой модели взаимодействия участников учебного процесса

Нахождение определенного в параграфе 1.4 оптимального варианта ИЭОС  $optimum(B)$  должно достигаться через реализацию в модели взаимодействия участников учебного процесса функций индивидуального уровня, удовлетворяющих следующим требованиям:

- 1) описывать формирование и корректировку ИОТ с учетом недетерминированности прохождения обучающимся индивидуального учебного плана вследствие влияния деструктивных факторов;
  - 2) учитывать распределение времени участников учебного процесса с возможностью увеличения времени контактной работы обучающегося и преподавателя в условиях ограниченного времени преподавателя;

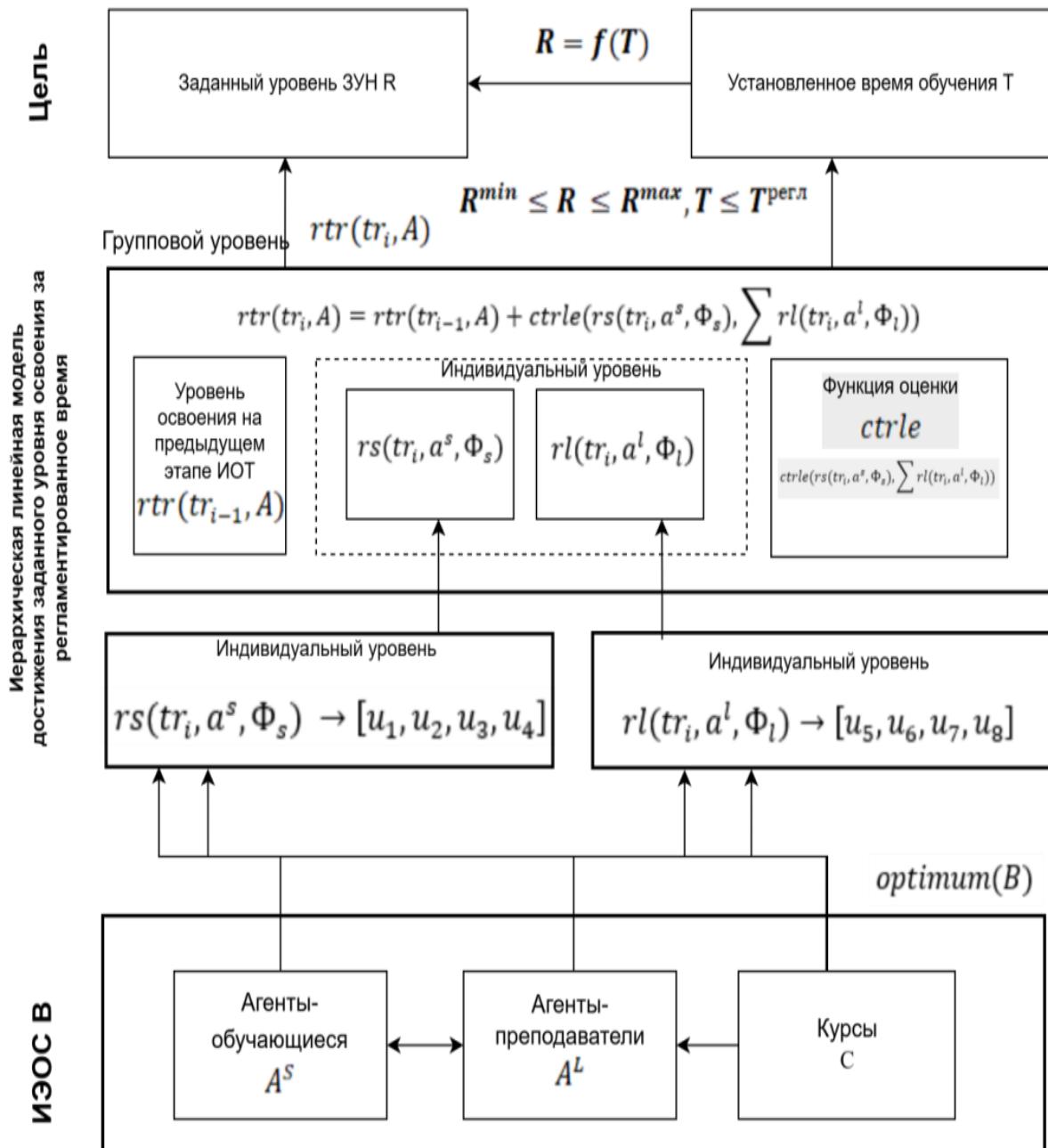
3) формализовать минимизацию ошибок обучающегося и ошибок оценки в условиях ограниченного времени самостоятельной и контактной работы и необходимости достижения заданного уровня освоения курса в установленные сроки.

Для выполнения указанных требований необходимо обосновать состав и структуру модели взаимодействия участников учебного процесса и описать ее важнейшие компоненты.

## **2.2 Обоснование состава и структуры модели взаимодействия участников учебного процесса**

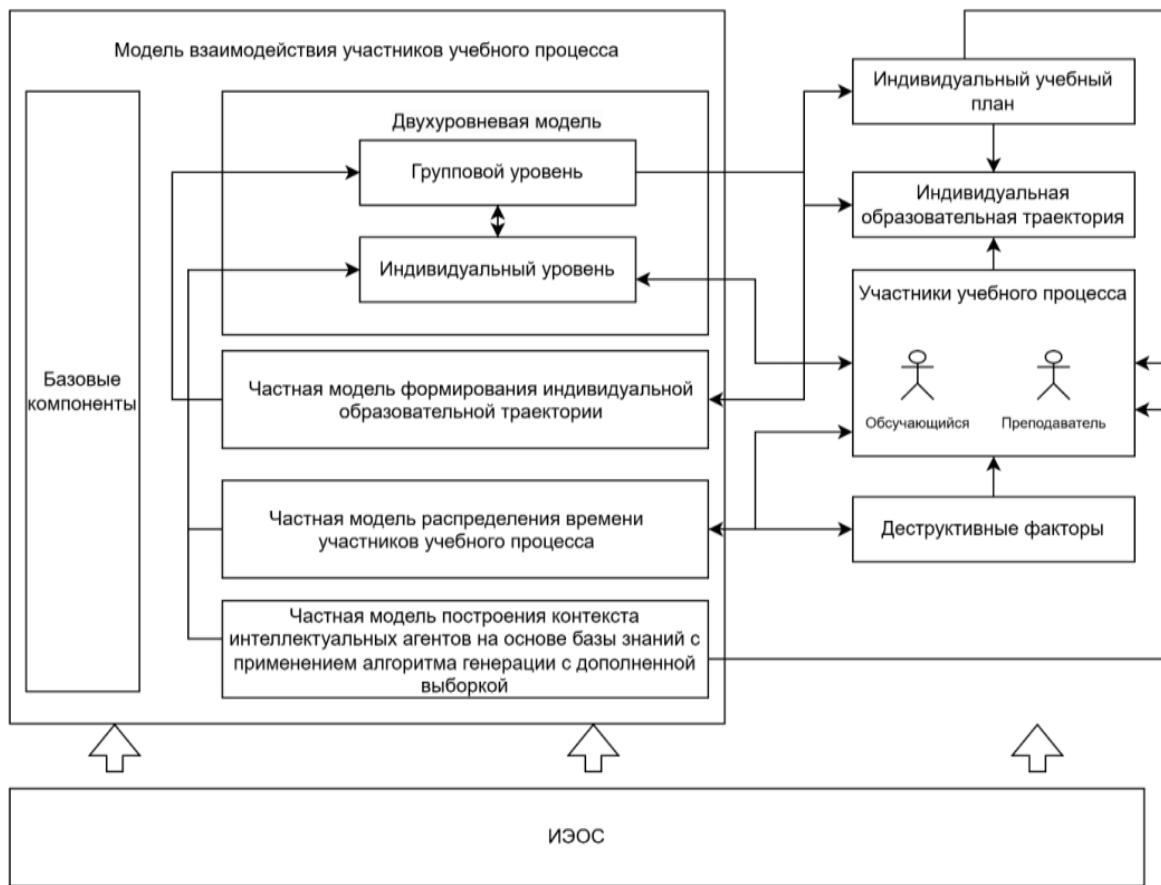
Проведенные исследования позволили сформировать гипотезу, что оптимальный вариант ИЭОС *optimit (B)* может быть реализован через описание функций индивидуального уровня двухуровневой модели взаимодействия участников учебного процесса. В этом случае связь индивидуального и группового уровня модели взаимодействия с компонентами ИЭОС *B* может быть представлена в виде структурно-логической схемы на рисунке 5 и включает в своем составе как базовые компоненты, отражающие объекты и множества, которые используются во всех частных моделях и методике оперативного формирования ИОТ, так и три частных модели, описывающих выполнение требований к модели взаимодействия на индивидуальном групповом уровнях.

В общем виде состав и структура двухуровневой модели взаимодействия участников учебного процесса представлены на рисунке 6. Базовые компоненты используются на всех уровнях модели, а также в частных моделях, которые уточняют различные аспекты функционирования системы «Преподаватель – Обучающийся» и позволяют описать особенности модели взаимодействия участников учебного процесса, отличающие ее от ранее предложенных решений.



Источник: составлено автором.

Рисунок 5 – Структурно-логическая схема связи двухуровневой модели взаимодействия участников учебного процесса и ИЭОС



Источник: составлено автором.

Рисунок 6 – Структурно-логическая схема модели взаимодействия участников учебного процесса

Детальное описание двухуровневой модели взаимодействия участников учебного процесса в части описания функционирования ее базовых компонентов, используемых как на индивидуальном, так и на групповом уровнях представлено ниже.

**Формирование состава базовых компонентов модели взаимодействия участников учебного процесса.** Базовые компоненты, отражающие объекты и множества, которые используются во всех частных моделях и методике оперативного формирования ИОТ в формальном виде могут быть описаны как шестерка (9)

$$\langle A, c, Pl, Tr, Kb, \Phi \rangle, \quad (9)$$

где  $A$  – множество участников учебного процесса;

$c$  – выбранный для прохождения курс,  $c \in C$ ;

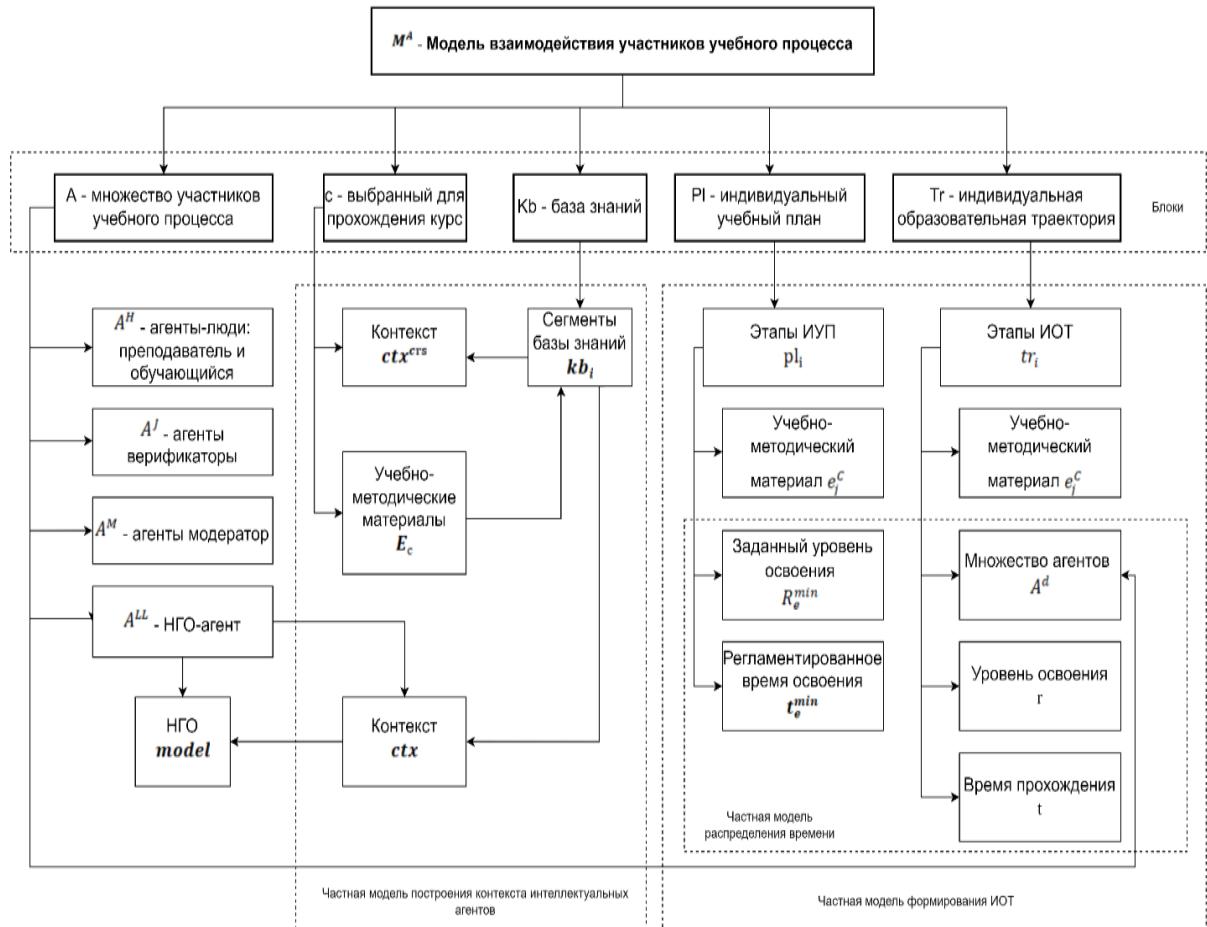
$Pl$  – индивидуальный учебный план;

$Tr$  – индивидуальная образовательная траектория;

$Kb$  – база знаний интеллектуальных агентов;

$\Phi$  – множество деструктивных факторов.

Базовые компоненты  $A, c, Pl, Tr, Kb$  модели в графическом виде представлены на рисунке 7 и позволяют определить инструментарий, обеспечивающий выполнение требований, приведенных в параграфе 2.1.



Источник: составлено автором.

Рисунок 7 – Структурная схема базовых компонентов модели взаимодействия

Для детализации того, как будут выполнены требования, приведенные в параграфе 2.1 опишем каждый базовый компонент модели подробнее.

### Базовый компонент «Множество участников учебного процесса».

Множество участников учебного процесса  $A$  описывается в формулах (10), (11) и (12)

$$A = \{a_1^L\} \cup \{a_1^S\} \cup A^J \cup \{a_0^M\} \cup A^{LL}, \quad (10)$$

$$A^{LL} = \{a_1^{LL}, a_2^{LL} \dots, a_n^{LL}\}, \quad (11)$$

$$A^J = \{a_1^J, a_2^J \dots, a_n^J\}, \quad (12)$$

где  $a_1^L \in A^L$  – преподаватель. В модели сделано допущение о том, что один курс ведет только один преподаватель.

$a_1^S \in A^S$  – обучающийся. Модель применяется к каждому обучающемуся отдельно, поскольку процесс достижения заданного уровня ЗУН за регламентированное время является индивидуальным.

$A^J$  – множество агентов-верификаторов. Агентами-верификаторами могут быть программные модули, способные выполнять функции оценки *ctrlle*. Множество агентов-верификаторов включает не менее одного агента-верификатора  $a_i^{JO} \in A^J$  для формальной проверки ответов и агента-верификатора  $a_i^{JB} \in A^J$ , способного выполнить программный код и запустить тесты.

$a_0^M$  – Агент-модератор, обрабатывающий и маршрутизирующий все естественно-языковые сообщения, которыми обмениваются агенты.

$A^{LL}$  – множество агентов, являющихся большими языковыми моделями, такое, что в нем обязательно наличие агента-преподавателя  $a_r^{ll} = a_i^{LT} \in A^{LL}$ , агента-практика  $a_i^{LP} \in A^{LL}$  и агента-инспектора  $a_i^{LV} \in A^{LL}$ . Каждый агент  $a_i^L$  описывается следующей формулой (13)

$$a_i^L = <\pi, mdl, ctx>, \quad (13)$$

где  $\pi$  – стратегия агента, которая заключается в стремлении агента к выполнению критерия оптимальности ИОТ, достижение которого обуславливается действиями участников учебного процесса на

индивидуальном и групповом уровнях, а также их последующей оценкой;

*model* – нейронная сеть глубокого обучения, на которой основан агент. Выбор НГО зависит от процедуры принятия решений о пригодности модели для реализации роли агента;

*ctx* – контекст, под которым в настоящей работе понимается совокупность ситуационных данных, предоставляемых агенту в рамках входного запроса, требующихся для точной интерпретации намерений пользователя, генерации релевантных ответов и поддержания связности диалога. Подробное описание структуры контекста *ctx* приведено в частной модели построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой в параграфе 2.3.

Представленное описание множества участников учебного процесса *A* позволяет применить так называемый многоагентный подход, при котором ИЭОС *B* рассматривается как система, согласно С. Расселу и П. Норвигу, состоящая из множества взаимодействующих агентов, под которыми в настоящей диссертации понимаются участники учебного процесса в совокупности с интеллектуальными агентами ИЭОС *B*. Каждый агент «оценивает собственное состояние, состояние других объектов и окружающей среды для выполнения своих действий, включая прогнозирование и планирование, которые максимизируют успешность, в том числе при неожиданном изменении оцениваемых состояний, достижения своих целей». Таким образом, формализуя участников учебного процесса, можно удовлетворить требования, приведенные в параграфе 2.1, а именно учесть недетерминированность прохождения ИУП при формировании ИОТ, учесть распределение времени и максимизировать время контактной работы и задать

критерии минимизации ошибок для достижения заданного уровня ЗУН за регламентированный срок.

### **Базовый компонент «Выбранный для прохождения курс».**

Выбранный для прохождения курс  $c \in C$  описывается формулой (14)

$$c = \langle R^{min}, R^{max}, T^{max}, T^{min}, Q, E, K \rangle, \quad (14)$$

где  $R^{min}$  – удовлетворительная оценка уровня ЗУН в баллах согласно образовательной программе курса;

$R^{max}$  – максимальная оценка уровня ЗУН в баллах согласно образовательной программе курса;

$T^{max}$  – максимально допустимое время изучения курса (в часах), включающее регламентированное время изучения курса согласно образовательной программе дисциплины и время, затраченное преподавателем на работу по курсу;

$T^{min}$  – минимальное время (в часах) изучения курса, может быть оценено как время контактной работы  $t^{Конт}$ ;

$Q$  – множество тем, содержащихся в выбранном курсе  $c$ ;

$E$  – множество учебно-методических материалов курса  $c$ , под которыми в настоящей диссертации понимается хранящаяся в базе данных ИЭОС текстовая или графическая информация, создаваемая и/или используемая как средство обучения по выбранному курсу  $c$ , например лекций, пособий, дидактических материалов.

Каждый учебно-методический материал  $e_i$  описывается формулой (15)

$$e_i = \langle q_k, data \rangle \quad (15)$$

где  $q_k$  – тема, к которой относится данный учебно-методический материал  $e_i$ .

$data$  – текстовая или графическая информация учебно-методического материала. Например, для ИЭОС такой информацией может быть

веб-страница, текстовые поля в базе данных, текстовый документ и другие.

$K$  – множество оценочных средств курса  $C$ , под которыми в настоящей диссертации понимается совокупность контролирующих материалов (типовые опросы, задачи, тесты и так далее), используемых при проведении текущего контроля уровня ЗУН по теме  $q_k$ .

Каждый элемент множества оценочных средств описывается следующей формулой (16)

$$k_i = \langle q_k, data, type \rangle, \quad (16)$$

где  $q_k$  – тема, к которой относится оценочное средство  $k_i$ .

$data$  – содержание оценочного материала, например список вопросов или задач. Необходимо заметить, что в настоящей диссертации делается допущение о том, что в ИЭОС содержатся правильные ответы на вопросы и проверочные данные для задач.

$type$  – тип оценочного средства. В настоящей диссертации выделяются три типа оценочных средств:  $type_1$  – тестовое задание,  $type_2$  – задание на составление программы,  $type_3$  – опрос.

Для демонстрации эффективности представленной формализации приведем в виде пример из опыта курса «Структуры и алгоритмы обработки данных». Приведем некоторые элементы множества тем  $Q$  курса:

- 1)  $q_1$  – «Понятие алгоритма и структуры данных. Свойства алгоритма. Этапы построения алгоритмов»;
- 2)  $q_2$  – «Классификация структур данных. Статические, динамические и файловые структуры данных. Линейные и нелинейные структуры данных»;
- 3)  $q_3$  – «Типовые структуры описания абстрактных данных. Массив, список, стек, очередь»;

По теме  $q_3$  «Типовые структуры описания абстрактных данных. Массив, список, стек, очередь» в ИЭОС содержатся следующие учебно-методические материалы  $e_i$ :

- 1)  $e_1(q_3)$  – текст лекции «Типовые структуры описания абстрактных данных. Массив, список, стек, очередь»;
- 2)  $e_2(q_3)$  – текст «Краткий методический материал для выполнения лабораторной работы «Массивы».

Множество  $K_c$  по теме  $q_4$  «Типовые структуры описания абстрактных данных. Массив, список, стек, очередь» в ИЭОС включает следующие оценочные материалы:

- 1)  $k_1(q_3)$  – сборник вопросов в части использования массивов, списков, стеков и очередей;
- 2)  $k_2(q_3)$  – лабораторная работа «Массивы», содержащая 26 вариантов.

Формализация структуры курса позволяет использовать его для формирования базы знаний  $Kb$ , а также позволяет описать индивидуальный учебный план в рамках курса.

**Базовый компонент «Индивидуальный учебный план»**  
Исследования, проведенные в диссертации, позволили описать ИУП  $Pl$  как связный ориентированный граф, описываемый формулой (17)

$$Pl = \langle Vpl, Epl \rangle, \quad (17)$$

где  $Vpl$  – множество вершин графа  $Pl$ , в котором каждая вершина является этапом ИУП и описывается формулой (18)

$$pl_i = \langle E', rtr^{min}, rtr^{max}, t_e^{\text{пегл}} \rangle, \quad (18)$$

где  $E' \subseteq E_c$  – учебно-методические материалы курса  $c$ , изучаемые на данном этапе ИУП;

$rtr^{min}$  – удовлетворительная оценка уровня ЗУН в баллах для данного этапа ИУП;

$rtr^{max}$  – максимальная оценка уровня ЗУН для данного этапа ИУП;

$t_e^{\text{регл}}$  – заданное время на прохождение  $i$ -го этапа ИУП, которое описывается формулой (19)

$$t_e^{\text{регл}} = \langle t_e^{\text{Конт}}, t_e^{\text{CPC}}, t_e^{\Delta}, t_e^{\text{СРП}} \rangle, \quad (19)$$

где  $t_e^{\text{Конт}}$  – время контактной работы обучающегося и преподавателя;

$t_e^{\text{CPC}}$  – время самостоятельно работы обучающегося;

$t_e^{\Delta}$  – время, реагирования на деструктивные факторы;

$t_e^{\text{СРП}}$  – время самостоятельной работы преподавателя;

$Epl$  – множество ребер графа  $Pl$ , в котором каждое ребро является переходом от одного этапа ИУП к другому и описывается следующей формулой (20)

$$epl_k = \langle pl_i, pl_j, R^{\text{пер}}, p \rangle, \quad (20)$$

где  $pl_i$  – этап ИУП, являющийся начальной вершиной ребра;

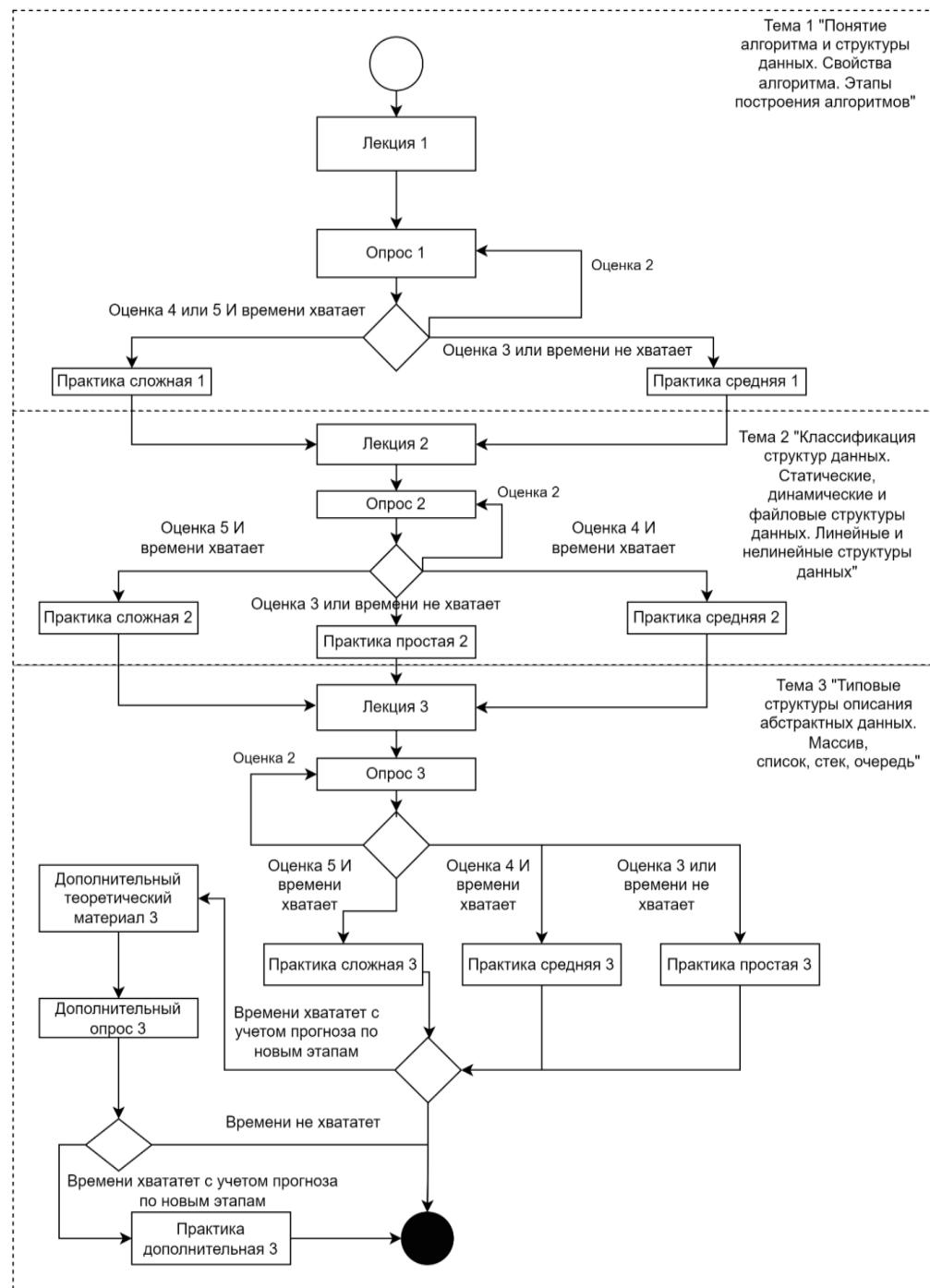
$pl_j$  – этап ИУП, являющийся конечной вершиной ребра;

$R^{\text{пер}}$  – интервал значений балльной шкалы, соответствующих данному ребру. Переход от этапа  $pl_i$  к этапу  $pl_j$  происходит, если этап  $pl_i$  завершен с баллом  $r_{ij} \in R^{\text{пер}}$ , который удовлетворяет условию соблюдения заданного уровня ЗУН  $rtr_i^{min} \leq r_{ij} \leq rtr_i^{max}$ ;

$p$  – вероятность перехода по данному ребру.

Граф  $Pl$  должен содержать в себе не менее одной цепи (маршрут без повторяющихся ребер), соединяющей начальную и конечную вершины, то есть начальный и конечный этапы ИУП. Примерный вид ИУП в виде графа для курса «Структуры и алгоритмы обработки данных» по перечисленным выше темам  $q_1, q_2, q_3$  представлен на рисунке 8.

Представление ИУП в виде графа  $Pl = \langle Vpl, Epl \rangle$  обеспечивает возможность построения ИОТ путем нахождения кратчайшего пути на ориентированном графе и возможность применения МППР при выборе переходов  $Epl$  между этапами-вершинами  $Vpl$ .



Источник: составлено автором.

Рисунок 8 – Примерный вид ИУП для курса  
«Структуры и алгоритмы обработки данных»

**Базовый компонент «Индивидуальная образовательная траектория».** Задачей компонента является формализация индивидуальной образовательной траектории как пути на ориентированном графе  $Pl$ . Путем на ориентированном графе называется маршрут с различными вершинами, что позволяет описать ИОТ  $Tr$  формулами (21) и (22)

$$Tr = \{tr_1, tr_2, \dots, tr_m\}, \quad (21)$$

$$tr_i = \langle pl_k, A^d, D, r_{ij}, t^{\Phi\text{акт}} \rangle, \quad (22)$$

где  $pl_k$  – этап ИУП;

$A^d$  – множество агентов, задействованных в прохождении этапа, в том числе агент-обучающийся  $a_j^{HS}$ .

Множество  $A^d$  отвечает условию (23)

$$A^d \ni a_j^{HS} \wedge A^d \ni a_0^M \wedge A^d \supseteq \{a_i^{LT}, a_i^{LP}, a_i^{LV}\} \wedge A^d \supseteq \{a_i^{JO}, a_i^{JB}\} \wedge A^d \subseteq A, \quad (23)$$

$D$  – множество диалогов(взаимодействий), формируется агентами при прохождении ИОТ  $tr_i$  и описывается формулами (24) и (25)

$$D = \{d_1, d_2, \dots, d_n\}, \quad (24)$$

$$d_i = \langle A_{di}^D, Txt, dr \rangle. \quad (25)$$

Диалогом считается любое обсуждение между агентами, в том числе обсуждение решения задач или контрольных примеров. Пример одного из диалогов  $d_1$ , проведенных между обучающимся и интеллектуальным агентом приведен на рисунке 9. Полный диалог приведен в приложении Е.

Твой код был запущен с следующими тестовыми данными:  
`str1 = "Тестовая"`  
`str2 = "Строка"`  
Ожидается ответ: Ттсооа  
Используй их и исправь код в случае ошибок.

### Ответ

К сожалению, предоставленный вами код не работает корректно с данными тестовыми значениями. После исправления ошибок и оптимизации, вот рабочий вариант функции:

```

1 public static String splitUpAndMerge(String str1, String str2) {
2     int min = Math.min(str1.length(), str2.length());
3     String s3 = "";
4
5     // Используем цикл for для перебора символов в каждой строке
6     for (int i = 0; i < min; i++) {
7         // Проверяем, является ли индекс четным, чтобы добавить символ из str1
8         if (i % 2 == 0) {
9             s3 += str1.charAt(i);
10        } else { // Если индекс нечетный, добавляем символ из str2
11            s3 += str2.charAt(i);
12        }
13    }
14
15    return s3;
16 }
```

Когда мы запускаем этот код с тестовыми данными `str1 = "Тестовая"` и `str2 = "Строка"`, результатом будет "Ттсооа", как и ожидалось.

Источник: составлено автором.

Рисунок 9 – Пример диалога между обучающимся и интеллектуальным агентом

Каждый диалог  $d_i$  состоит может быть описан формулой (26)

$$d_i = \langle A_{di}^D, Txt_i, dr \rangle, \quad (26)$$

где  $A_{di}^D$  – агенты, участвующие в диалоге. В примере, приведенном на рисунке 9  $A_{d1}^D = \{\text{Обучающийся}, \text{Агент-разработчик}\}$ .

$Txt_i$  – множество сообщений (реплик) диалога. В таком формализме первое сообщение из приведенного на рисунке 9 примера будет выглядеть следующим образом:  $txt_1 \in Txt_{d1} = \langle \text{Твой код был запущен с следующими тестовыми данными:} str1 = \langle \text{«Тестовая}} \rangle str2 = \langle \text{«Строка}} \rangle \text{ Ожидается ответ: Используй их и исправь код в случае ошибок}} \rangle$ .

$dr$  – балл, полученный обучающимся за диалог, если диалог требовал оценивания (задача или контрольный пример), при этом  $dr \in R_e^{\text{регл}}$ . Приведенный на рисунке 9 диалог не требовал оценивания, поэтому балл  $dr$  будет пуст.

$r_{ij} \in R_e^{\text{регл}}$  – балл, полученный за прохождение этапа ИУП  $pl_k$ , соответствующего этапу ИОТ  $tr_i$ . Функции, описывающие определение балла  $r_{ij}$  подробно описаны выше.

$t^{\text{Факт}}$  – фактически затраченное время на прохождение  $i$ -го этапа ИУП, которое описывается формулой (27)

$$t^{\text{Факт}} = \langle t_{\text{Факт}}^{\text{Конт}}, t_{\text{Факт}}^{\text{СРС}}, t_{\text{Факт}}^{\Delta}, t_{\text{Факт}}^{\text{СРП}} \rangle, \quad (27)$$

где  $t_{\text{Факт}}^{\text{Конт}}$  – фактическое время контактной работы обучающегося и преподавателя;

$t_{\text{Факт}}^{\text{СРС}}$  – фактическое время самостоятельно работы обучающегося;

$t_{\text{Факт}}^{\Delta}$  – фактическое время, затрачиваемое на реагирование на деструктивные факторы;

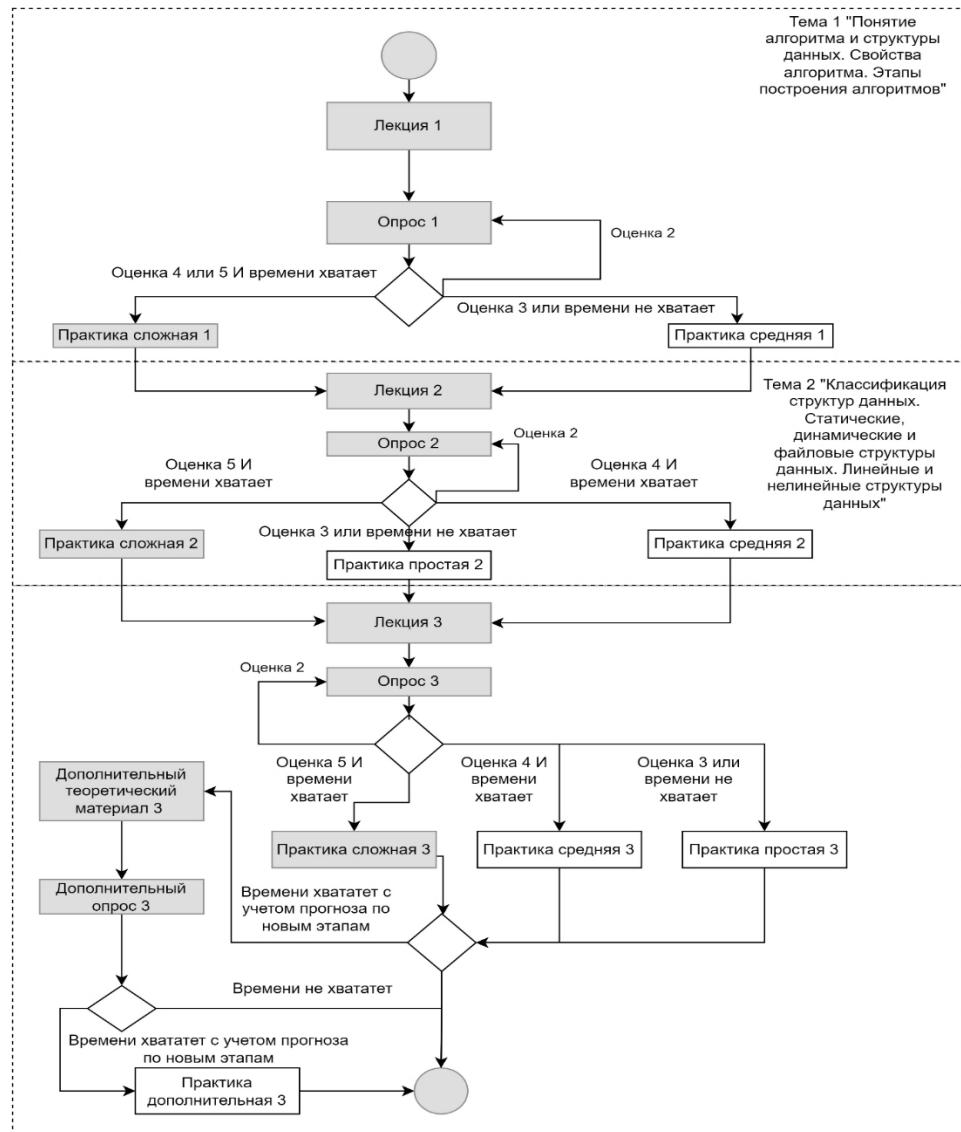
$t_{\text{Факт}}^{\text{СРП}}$  – фактическое время самостоятельной работы преподавателя.

ИОТ будет удовлетворять цели настоящей работы в части обеспечения достижения требуемого уровня знаний, умений и навыков за установленное время при обучении программированию в интеллектуальной электронной образовательной системе, если для будет выполнено условие, описываемое формулой (28)

$$R^{\min} \leq \sum_{i=1}^{|Tr|} r_{ij} \leq R^{\max}, \sum_{i=1}^{|Tr|} t^{\text{Факт}} \leq T^{\max}. \quad (28)$$

Примерный вид ИОТ в виде маршрута на графике  $Pl$  для курса «Структуры и алгоритмы обработки данных» по перечисленным выше темам  $q_1, q_2, q_3$  и пройденного обучающимся с высокой скоростью и успеваемостью,

представлен на рисунке 10. Пройденные этапы выделены цветом. Например, при переходе от опроса по теме «Понятие алгоритма и структуры данных. Свойства алгоритма. Этапы построения алгоритмов» к практике агент-верификатор принял решение о выборе сложной практики, поскольку ИУП предусмотрено время прохождения  $T^{\text{регл}} = 9,0625$ , а фактические значения  $rtr_2 = 9,05, T = 7,5$  были меньше.



Источник: составлено автором.  
Рисунок 10 – Примерный вид ИОТ для курса «Структуры и алгоритмы обработки данных»

Приведенное выше описание ИУП *Pl* и ИОТ *Tr* позволяет задать пространство состояний учебного процесса для применения МППР для

описания формирования и корректировки ИОТ и последующего синтеза методики.

#### **Базовый компонент «База знаний интеллектуальных агентов».**

База знаний  $Kb$  состоит из компетентностного сегмента  $Kb^A$  и сегмента общего назначения  $Kb^O$ .

$Kb^A$  – компетентностный сегмент базы знаний, содержащий знания в сфере обучения программированию. Компетентностный сегмент базы знаний формируется на основе учебно-методических материалов  $E_c$  курса с.

$Kb^O$  – сегмент базы знаний общего назначения. В качестве сегмента базы знаний общего назначения использована открытая база знаний DBpedia.

Базы знаний обеспечивает выполнение требований в части минимизации ошибок обучающегося и ошибок оценки путем построения контекстов  $ctx$  интеллектуальных агентов  $a_i^L$  для оптимизации стратегии агентов.

Таким образом база знаний становится источником для формирования контекста интеллектуальных агентов, необходимого для применения нейро-нечеткого подхода к формированию ИОТ  $Tr$  из ИУП  $Pl$ , для достижения заданного уровня ЗУН  $R$  за регламентированное время  $T$  с учетом деструктивных факторов.

#### **Базовый компонент «Множество деструктивных факторов».**

$\Phi$  – множество деструктивных факторов, состоящее из подмножеств  $\Phi_s$  – деструктивных факторов, влияющих на обучающегося и  $\Phi_l$  – деструктивных факторов, влияющих на преподавателя. Множество  $\Phi$  включает в себя деструктивные факторы, основанные на описанных в параграфе 1.1 факторах по Дж. Хэтти, подробное описание формирования множеств  $\Phi_s$  и  $\Phi_l$  приведено в параграфе 2.3.

После описания базовых компонентов перейдем к формированию иерархической структуры модели взаимодействия участников учебного процесса на основе базовых компонентов. Это в свою очередь дает

возможность выбрать оптимальный вариант ИЭОС *optimum (B)* и достигнуть цель работы.

**Формирование двухуровневой структуры модели взаимодействия участников учебного процесса.** В параграфе 2.1 была выдвинута гипотеза о том, что двухуровневая модель взаимодействия участников учебного процесса может при описании внутренних зависимостей использовать нейронные сети глубокого обучения. Это обусловлено тем, что анализ приведенных в главе 1 работ [96; 106; 110; 111; 113; 123; 127; 130] показывает невозможность с помощью существующих моделей (логистическая функция, модель Раша, экспоненциальная модель, Item Response Theory, иерархические линейные модели) в должной мере учесть влияние деструктивных факторов и описать недетерминированность учебного процесса.

Влияние деструктивных факторов и недетерминированности учебного процесса может быть учтено путем применения такого раздела «мягких вычислений» как нейронные сети. Применимость данного аппарата к задаче нахождения ИЭОС, в которой выполняется критерий оптимальности  $(R < R^{\min} \wedge T^{\max} - T \leq \varepsilon) \vee (R^{\min} < R \leq R^{\max} \wedge T^{\max} - T > \varepsilon)$  обусловлена тем, что любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике согласно теореме Fuzzy Approximation Theorem, доказанной Б. Коско в 1993 г.

Механизм нечетких выводов в своей основе имеет базу знаний, формируемую специалистами предметной области в виде совокупности нечетких предикатных правил. Однако, формирование базы знаний является трудоемким и долгим процессом. В связи с этим в последнее время появляются работы, в которых описано применение нейросетей глубокого обучения для реализации механизмов нечеткого вывода [38; 39; 42].

Существуют исследования [32; 94], которые показывают, что при оценивании обучающихся преподаватели применяют неформализованные процедуры, поэтому в настоящей диссертации предлагается использовать

НГО при вычислении функций, описывающих зависимость достижения заданного уровня ЗУН за регламентированное время в системе «Преподаватель – Обучающийся».

Взаимодействие участников учебного процесса описывается с применением так называемого агентного подхода [2], при котором участники учебного процесса – преподаватель и ученик – образуют два уровня иерархической модели:

- 1) индивидуальный уровень (преподаватель, обучающийся);
- 2) групповой уровень (взаимодействующие преподаватель и обучающийся).

Для достижения цели диссертации крайне важно, чтобы индивидуальный и групповой уровень были описаны с применением базовых компонентов, введенных в настоящей главе. С этой целью представим более детально указанные компоненты.

**Индивидуальный уровень.** Индивидуальный уровень модели включает функции  $rs(tr_i, a^s, \Phi_s)$  и  $rl(tr_i, a^l, \Phi_l)$ .

Функция  $rs(tr_i, a^s, \Phi_s)$  описывает зависимость результата на i-ом этапе индивидуальной образовательной траектории от действий обучающегося. На вход функции подаются данные, собранные при взаимодействии обучающегося, преподавателя и интеллектуальных агентов. Функция выполняет преобразование этих данных путем обращения к агенту-верификатору  $a^J$ . Агент-верификатор формирует вектор из 4 выходных текстовых концептуальных выражений в виде функции (29)

$$rs(tr_i, a^s, \Phi_s) \rightarrow [u_1, u_2, u_3, u_4], \quad (29)$$

где  $tr_i$  – i-й этап ИОТ.

$a^s$  – обучающийся.

$\Phi_s$  – множество деструктивных факторов, действующих на обучающегося.

$u_1$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, превышает ли  $t_{\text{Факт}}^{\text{CPC}}$  заданное в ИУП для данного этапа  $pl_k$  значение  $t_e^{\text{CPC}}$ . Текстовое концептуальное выражение  $u_1$  может принимать следующие значения: «Существенно отстает», «Отстает», «Успевает», «Опережает», «Существенно опережает».

$u_2$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, превышает ли  $t_{\text{Факт}}^{\text{Конт}}$  заданное в ИУП для данного этапа  $pl_k$  значение  $t_e^{\text{Конт}}$ . Текстовое концептуальное выражение  $u_2$  может принимать следующие значения: «Существенно отстает», «Отстает», «Успевает», «Опережает», «Существенно опережает».

Для вычисления текстовых концептуальных выражений  $u_1$  и  $u_2$  при оценке этапов курса «Структуры и алгоритмы обработки данных» агенту-верификатору были даны рекомендации придерживаться оценок, приведенных в таблице 2.

Таблица 2 – Рекомендации по соответствующим значениям текстовых концептуальных выражений

Текстовое концептуальное выражение	Рекомендация
«Существенно отстает»	Фактическое время превышает плановое более чем на 30%
«Отстает»	Фактическое время превышает плановое более чем на 10%, но менее, чем на 30%
«Успевает»	Фактическое время превышает плановое менее чем на 10% или меньше планового не более, чем на 10%
«Опережает»	Фактическое время меньше планового более чем на 10%, но менее, чем на 30%
«Существенно опережает»	Фактическое время меньше планового более чем на 30%

Источник: составлено автором.

Для вычисления текстового концептуального выражения  $u_3$  агенту-верификатору дается следующая рекомендация:

$u_3$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, как агент-верификатор  $a^{Jl}$  оценивает все диалоги  $D$  на этапе  $tr_i$  с учетом возможных выставленных баллов  $dr$ . Текстовое концептуальное

выражение  $u_3$  может принимать значения, образуемые путем построения пары из термов: «Неудовлетворительно», «Удовлетворительно», «Хорошо», «Отлично» и термов «Менее чем», «Чуть менее чем», «Чуть более чем», «Более чем».

$u_4$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, какие деструктивные факторы, относящиеся к  $\Phi_s$ , повлияли на прохождение этапа  $tr_i$  на основании оценки диалога  $D$  и насколько сильным было влияние. Текстовое концептуальное выражение  $u_4$  может принимать значения, образованные путем составления пар из выявленных деструктивных факторов  $\varphi_i$  и следующих оценок: «Незначительно», «Средне», «Значительно».

Функция  $rl(tr_i, a^l, \Phi_l)$  описывает зависимость результата на  $i$ -ом этапе индивидуальной образовательной траектории от действий преподавателя. На вход функции подаются данные, собранные при взаимодействии обучающегося, преподавателя и интеллектуальных агентов. Функция выполняет преобразование этих данных путем обращения к агенту-верификатору  $a^{Jl}$ . Агент-верификатор формирует вектор из 4 выходных текстовых концептуальных выражений в виде функции (30)

$$rl(tr_i, a^l, \Phi_l) \rightarrow [u_5, u_6, u_7, u_8], \quad (30)$$

где  $tr_i$  –  $i$ -й этап ИОТ.

$a^l$  – преподаватель.

$\Phi_l$  – множество деструктивных факторов, действующих на преподавателя.

$u_5$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, превышает ли  $t_{\text{Факт}}^{\text{СРП}}$  заданное в ИУП для данного этапа  $pl_k$  значение  $t_e^{\text{СРП}}$ . Текстовое концептуальное выражение  $u_1$  может принимать следующие значения: «Существенно отстает», «Отстает», «Успевает», «Опережает», «Существенно опережает».

$u_6$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, превышает ли  $t_{\text{Факт}}^{\text{Конт}}$  заданное в ИУП для данного этапа  $pl_k$  значение  $t_e^{\text{Конт}}$ . Текстовое концептуальное выражение  $u_2$  может принимать следующие значения: «Существенно отстает», «Отстает», «Успевает», «Опережает», «Существенно опережает».

$u_7$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, как агент-верификатор  $a^J$  оценивает все диалоги  $D$  на этапе  $tr_i$ . Текстовое концептуальное выражение  $u_3$  может принимать значения, образуемые путем построения пары из термов: «Неудовлетворительно», «Удовлетворительно», «Хорошо», «Отлично» и термов «Менее чем», «Чуть менее чем», «Чуть более чем», «Более чем».

$u_8$  – выходное текстовое концептуальное выражение, являющееся ответом на вопрос, какие деструктивные факторы, относящиеся к  $\Phi_l$ , повлияли на прохождение этапа  $tr_i$  на основании оценки диалога  $D$  и насколько сильным было влияние. Текстовое концептуальное выражение  $u_4$  может принимать значения, образованные путем составления пар из выявленных деструктивных факторов  $\varphi_i$  и следующих оценок: «Незначительно», «Средне», «Значительно».

Такое представление функций индивидуального уровня позволяет показать прохождение текущего этапа ИУП  $tr_i$  как взаимодействие преподавателя и обучающегося по заданному учебно-методическому материалу  $e_i$ , в условиях ограниченного времени, контактной и самостоятельной работы при влиянии деструктивных факторов.

**Групповой уровень.** Групповой уровень модели описывается функцией (31)

$$rtr(tr_i, A) = rtr(tr_{i-1}, A) + ctrl(rs(tr_i, a^s, \Phi_s), \sum rl(tr_i, a^l, \Phi_l)), \quad (31)$$

где  $tr_i$  – i-й этап ИОТ;

$A$  – множество участников учебного процесса, такое, что  $a^s \in A, a^l \in A$ ;

$rtr(tr_{i-1}, A)$  – значение функции на предыдущем этапе индивидуальной образовательной траектории;

$ctrlle(rs(tr_i, a^s, \Phi_s), \sum rl(tr_i, a^l, \Phi_l))$  – функция, которая получает на вход векторы концептуальных текстовых переменных, передает их агенту-верификатору для выдачи оценки в баллах в от 1 до максимального уровня оценки данного этапа  $rtr^{max}$  индивидуальной образовательной траектории.

Применение нейро-нечетких систем для оценки компетенций ранее уже описывалось в источниках [17; 47; 63; 67; 84; 134], однако подход, при котором оценка осуществляется агентом, созданным на основе большой языковой модели, ранее не предлагался.

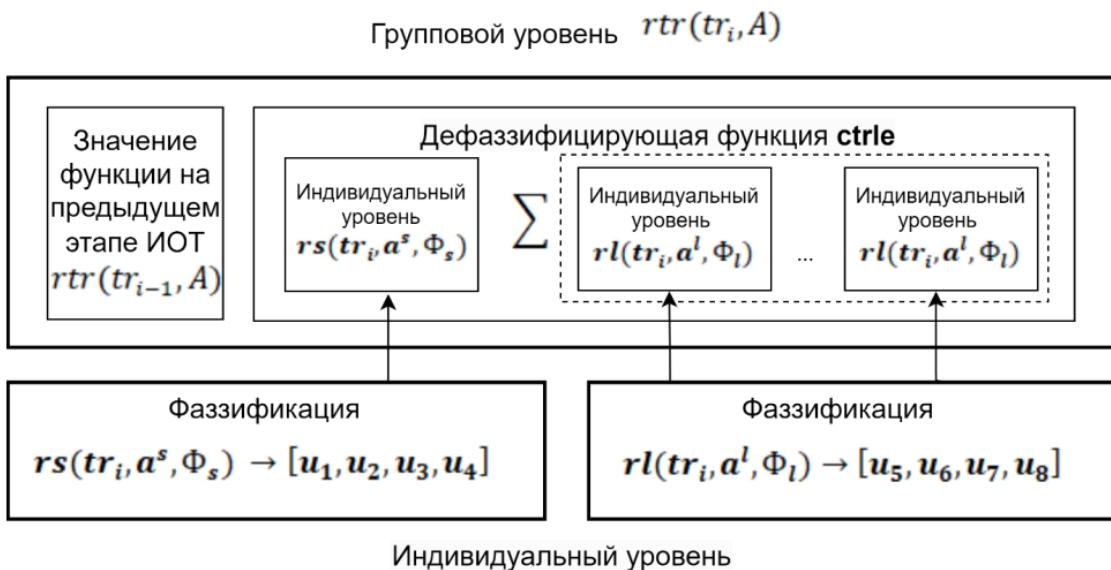
Представление в виде двух уровней приведено на рисунке 11 и позволяет учесть индивидуальные аспекты и аспекты взаимодействия «Обучающийся – Преподаватель». Суммирование значений функций второго уровня  $rl$  отражает тот факт, что на достижение заданного уровня ЗУН одного обучающегося могут влиять несколько преподавателей, например при разделении лекций и практик, ведении практик по разным темам разными преподавателями или в случае замены преподавателя.

Достижение заданного уровня ЗУН, учитываемое в ИЛМ путем вычисления  $rtr(tr_i, A)$ , может быть выражено формулой (32)

$$R = rtr(tr_n, A), \quad (32)$$

где  $rtr(tr_n, A)$  – значение уровня освоения на завершающем этапе индивидуальной образовательной траектории;

$N$  – общее количество этапов индивидуальной образовательной траектории.



Источник: составлено автором.

Рисунок 11 – Двухуровневая структура модели взаимодействия участников учебного процесса

Приведенная выше детализация двух уровней модели взаимодействия позволяет подтвердить выдвинутую в параграфе 2.1 гипотезу о том, что модель, решающая задачу построения аналитической зависимости для синтеза методики формирования индивидуальной образовательной траектории, должна обеспечивать проведение исследований в части использования для описания зависимостей предложенной Л. Заде концепции «мягких вычислений» и формирования двух уровней иерархии: индивидуального и группового. Однако для оптимизации функции  $optimum(B)$  необходимо описать, как модель взаимодействия выполняет требования из упомянутого параграфа. Для этого необходимо описать частные модели в составе модели взаимодействия участников учебного процесса.

Таким образом по результатам представления участников учебного процесса, как на индивидуальном, так и на групповом уровне, имитационными моделями с функциями интеллектуальных агентов сформирована двухуровневая зависимость индивидуальной образовательной траектории от текущих оценок показателей восприятия и реакции обучаемого.

### **2.3 Описание частных моделей, входящих в состав модели взаимодействия участников учебного процесса**

Модель взаимодействия участников учебного процесса включается три частные модели:

- частная модель формирования индивидуальной образовательной траектории;
- частная модель распределения времени участников учебного процесса;
- частная модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

Частные модели позволяют описать достижение заданного уровня ЗУН за регламентированное время с использованием описанных выше базовых компонентов, которые описаны.

Частная модель формирования индивидуальной образовательной траектории описывает с использованием формализма марковского процесса принятия решений параметры ИОТ с учетом результатов прогноза показателей восприятия и реакции обучаемых на изменения учебной программы по данным сформированной нейросети глубокого обучения.

Частная модель распределения времени участников учебного процесса позволяет учесть текущие показатели прохождении учебной программы для своевременной корректировки следующего этапа ИОТ с учетом воздействия деструктивных факторов.

Частная модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой позволяет описать функции интеллектуальных агентов для взаимодействия и текущей оценки показателей восприятия и реакции обучаемого, а также корректировать ИОТ с применением НГО-агентов.

В совокупности двухуровневая модель взаимодействия и дополняющие ее частные модели позволяют сформировать зависимость достижения обучающимся требуемого уровня знаний, умений и навыков в установленные образовательным стандартом сроки и на ее основе оперативно корректировать ИОТ для определения рациональной этапности обучения и необходимого времени.

**Частная модель формирования индивидуальной образовательной траектории.** Частная модель формирования индивидуальной образовательной траектории предназначена для описания формирования и корректировки ИОТ. Для этого частная модель описывает формирование ИОТ как марковский процесс принятия решений, а именно как достижение цели через взаимодействие и последовательное принятие решений.

Из описания базовых компонентов модели следует, что действия участников учебного процесса описываются с применением агентного подхода, выбор которого основан на том, что НГО-агенты могут выполнять нейро-нечеткие вычисления. С. Рассел и П. Норвиг [2], определяют агента как «субъект, действующий в среде для достижения целей и воспринимающий окружающую среду через сенсоры, а также воздействующий на неё через эффекторы», что обеспечивает возможность динамического формирования ИОТ на основе текущих параметров функций  $rtr(tr_i, A)$ ,  $rs(tr_i, a^s, \Phi_s)$  и  $rl(tr_i, a^l, \Phi_l)$  для достижения заданного уровня ЗУН за регламентированное время. Следуя этому определению, можно полагать, что агенты  $a^s \in A, a^l \in A$  – обучающийся и преподаватель – воспринимают образовательное пространство (среду) через входные параметры функций  $rs(tr_i, a^s, \Phi_s)$  и  $rl(tr_i, a^l, \Phi_l)$  и воздействуют на нее, путем агрегации значений этих функций на втором уровне иерархической линейной модели, формируя из статического учебного плана динамическую ИОТ.

В разделе 2.2 индивидуальный учебный план был представлен как граф  $Pl$ , а индивидуальная образовательная траектория  $Tr$  как путь на этом графике.

При этом интеллектуальный агент  $a_i \in A^{LL}$  для построения цепи на графе должен последовательно принимать решения по переходу от одного этапа к другому на основе текущего уровня ЗУН и оставшегося на прохождение курса времени.

В таком случае время пребывания на различных этапах ИУП будет разным, в частности детерминированным (на разные виды обучения отводится разное время). Переход от одного этапа к другому осуществляется по следующей схеме: если этап обучения  $tr_i$  закончен и  $r_{ij} \geq R_e^{\text{регл}}$ , то происходит переход  $epl_k$  к очередному этапу  $pl_i$ , в противном случае предусмотрено дополнительное обучение, что соответствует возврату в одно из предыдущих состояний, поэтому, процесс перехода на графике  $Pl$  случаен и может быть описан марковской цепью. Учитывая эти особенности, для построения моделей целесообразно использовать полумарковский случайный процесс [67] и формализм марковского процесса принятия решений.

Множеством состояний МППР для модели  $M^A$  являются вершины графа  $Pl$ , для каждой вершины которого  $pl_i$  доступно действие  $act = epl_k$  заключающееся в переходе в инцидентную вершину  $pl_j$  и являющееся ребром графа  $Pl$ . При выполнении действия  $act$  осуществляется добавление нового  $tr_i$  в ИОТ  $Tr$ . Вероятность перехода описывается переходной функцией  $P(pl_j|pl_i, act)$ , при этом за каждый переход назначается вознаграждение  $r_{ij}$  то есть балл, полученный за прохождение этапа  $pl_i$ . Стратегия выбора действий  $\pi$  описывается стратегией агента  $a_i^L$ .

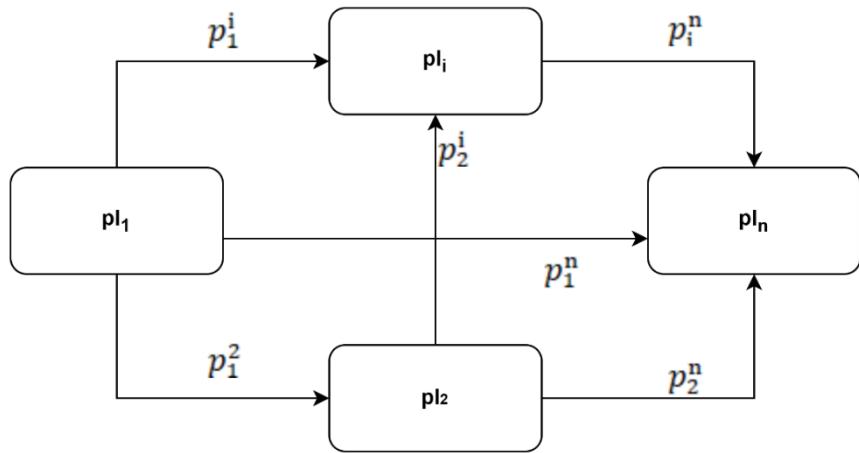
На основе графа  $Pl$  можно построить матрицу известных вероятностей перехода  $P(pl_j|pl_i, act)$ . В общем виде такая матрица может быть представлена формулой (33)

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{1i} & p_{1n} \\ p_{21} & p_{22} & p_{2i} & p_{2n} \\ p_{i1} & p_{i2} & p_{ii} & p_{in} \\ p_{n1} & p_{n2} & p_{ni} & p_{nn} \end{pmatrix}, \quad (33)$$

где  $p_{ij}$  – вероятность перехода из этапа ИУП  $pl_i$  к этапу  $pl_j$ .

Для неинцидентных вершин графа  $Pl$  вероятность перехода будет равна 0. Таким образом построение ИОТ формализуется как марковский процесс с частичной моделью.

Матрица  $P$  может быть визуализирована в виде графа, показанного на рисунке 12. Ребра, для которых  $p_{ij} = 0$ , удалены из графа.



Источник: составлено автором.  
Рисунок 12 – Визуализирующий матрицу  $P$  граф

Вершины приведенного на рисунке 12 графа соответствуют этапам ИУП, поэтому каждое ребро графа в терминах марковского процесса принятия решений является переходом вида  $(pl_i, act) \rightarrow pl_j$ . За каждый такой переход агент получает выраженное в балах «вознаграждение», которое рассчитывается с помощью функции полезности  $R(pl_i, act, pl_j)$ . С учетом того, что при прохождении каждого этапа ИОТ подсчитывается балл  $r_{ij}$  и фактическое время  $t^{\text{Факт}}$  можно определить функцию полезности  $R(pl_i, act, pl_j), pl_i \in tr_i$  используя функцию группового уровня  $rtr$ , описываемую формулой (34)

$$R(pl_i, act, pl_j) = rtr(tr_i, A) + E[rtr(pl_j, A)] + fw(pl_j, \varepsilon), \quad (34)$$

где  $rtr(tr_i, A)$  – фактическая оценка, полученная за пройденный этап ИОТ  $tr_i$ ;

$E[rtr(pl_j, A)]$  – математическое ожидание оценки на следующем этапе ИУП  $pl_j$ , включаемом в ИОТ;

$fw(pl_j, \varepsilon)$  – функция корректировки ожидаемой полезности с учетом суммарного времени, затраченного на прохождение ИОТ и порога риска  $\varepsilon$  – количества часов, после которого достижение заданного уровня ЗУН за регламентированное время считается невозможны.

Добавление функции  $fw(pl_j, \varepsilon)$  в расчет функции полезности обусловлен необходимостью учета, введенного в параграфе 1.4 критерия оптимальности, описываемого формулой (35)

$$R < R^{\min} \wedge T^{\max} - T \leq \varepsilon) \vee (R^{\min} < R \leq R^{\max} \wedge T^{\max} - T > \varepsilon). \quad (35)$$

Определение того, к какому этапу ИУП будет осуществлен переход, в формализме МППР зависит также от коэффициента дисконтирования  $\gamma \in [0,1]$ . Коэффициент дисконтирования определяет важность будущих вознаграждений по сравнению с текущим, поэтому в настоящей работе формула его определения на каждом этапе может описываться формулой (36)

$$\gamma = \begin{cases} 0,9, & \text{если } \sum R(pl_i, act, pl_j) < R^{\min} \\ 0,1, & \text{если } \sum R(pl_i, act, pl_j) \geq R^{\min}. \end{cases} \quad (36)$$

Подобный выбор коэффициента дисконтирования позволяет следовать стратегии, когда до достижения минимального уровня ЗУН  $R^{\min}$  важнее долгосрочное вознаграждение, выражаемое в достижении хотя бы минимального уровня за заданное время, а после достижения  $R^{\min}$  важнее максимизировать уровень ЗУН. Если говорить упрощенно, то при прохождении ИОТ важно достигнуть удовлетворительный уровень за отведенное время, а в оставшееся время (при условии его наличия) можно попробовать максимизировать этот уровень. Такая стратегия соответствует цели диссертации, однако в определённой мере стратегия может быть

скорректирована путем изменения коэффициента дисконтирования  $\gamma$ , например так, чтобы с начала формирования ИОТ ориентироваться на достижение  $R^{max}$ .

Действия агента описываются стратегией выбора  $\pi$ , которая заключается в максимизации ценности состояния, описываемого следующей формулой (37)

$$V(\pi, tr_i) = \max^{act} \sum_{pl_j} P(pl_j|pl_i, act) [R(pl_i, act, pl_j) + \gamma V(\pi, pl_j)]. \quad (37)$$

При этом функция полезности  $R(pl_i, act, pl_j)$  зависит от функции группового уровня  $rtr(tr_i, A)$  и функции корректировки  $fw(pl_j, \varepsilon)$ .

Детальное описание вычисления функции  $rtr(tr_i, A)$  приведено в этой главе выше, поэтому здесь остановимся на описании функции  $fw(pl_j, \varepsilon)$ , которая описывает зависимость выбора следующего этапа  $pl_{i+1}$  от текущего уровня ЗУН с учетом порога риска  $T^{регл} - T > \varepsilon$ . Учет порога риска можно описать уравнением (38)

$$T^{max} - T^{ост} = T^{max} - \sum_{i+1}^N t_e^{регл}, t_e^{регл} \in Pl^*, \quad (38)$$

где  $Pl^*$  – непройденная часть ИУП.

Для определения  $\sum_{i+1}^N t_e^{регл}, t_e^{регл} \in Pl^*$  в простейшем случае может быть применен алгоритм Левита, который позволяет найти кратчайшее расстояние между вершинами графа. Выразим  $T^{ост}$  через применение этого алгоритма к подграфу  $Pl^*$  в виде формулы (39)

$$T^{ост} = \min^{Lv} (\sum_{i+1}^N t_e^{регл}), t_e^{регл} \in Pl^*. \quad (39)$$

Таким образом функцию корректировки  $fw(pl_j, \varepsilon)$  можно описать формулой (40)

$$fw(pl_j, \varepsilon) = \frac{\varepsilon}{T^{ост}}. \quad (40)$$

Такая формализация отражает влияние функции корректировки ожидаемой полезности с учетом суммарного времени на функцию полезности  $R(pl_i, act, pl_j)$  при достижении значений времени Т близких к порогу риска. В таких случаях процесс прохождения ИОТ становится близок к порогу риска  $\varepsilon$ , а влияние функции корректировки ожидаемой полезности с учетом суммарного времени стремление агента выбрать кратчайший путь на графе  $Pl^*$  увеличиваются.

Например, при прохождении курса «Структуры и алгоритмы обработки данных» порог риска был установлен на уровне 30% от времени, предусмотренного учебным планом. Таким образом, стремление агента выбрать кратчайший путь на графике  $Pl^*$  увеличивалось, когда  $T^{ост}$  становилось менее 21 часа.

Найденный при вычислении функции  $fw(pl_j, \varepsilon)$  кратчайший путь является оптимальным по времени, но не учитывает достижение требуемого уровня освоения. Для расчета ценности этапа  $pl_j$  с учетом непройденных этапов ИУП может быть использован один из численных методов решения уравнения Беллмана, например метод обратного прохода, описываемый формулой (41)

$$V(pl_j) = \sum \pi(act|pl_j) \sum_{pl_i} P(pl_j|pl_i, act) [R(pl_i, act, pl_j) + \gamma V(\pi, pl_j)]. \quad (41)$$

В частной модели формирования ИОТ при решении марковского процесса учитывается основанная на контексте  $ctx$  способность интеллектуального агента  $a_i^L$  оценить входные векторы  $vrs$  и  $\sum vrl$  и распределение времени, входящего в векторы  $vrs$  и  $vrl$ , а именно  $t^{CPC}, t^{Конт}, t^{CPN}, t^{Конт}, t^D$  с учетом деструктивных факторов  $\Phi_s$  и  $\Phi_l$ . Это позволяет производить динамическую корректировку ИОТ с учетом текущего состояния, что необходимо для получения зависимости достижения заданного уровня ЗУН от регламентированного времени.

**Частная модель распределения времени участников учебного процесса.** Частная модель распределения времени участников учебного процесса предназначена для описания зависимости распределения времени между участниками учебного процесса от их действий в условиях влияния деструктивных факторов и позволяет выполнить требование к модели по учету распределения времени участников учебного процесса с возможностью увеличения времени контактной работы обучающегося и преподавателя в условиях ограниченного времени преподавателя.

Для этого в частной модели делается допущение, что оптимизация на индивидуальном уровне осуществляется путем увеличения времени самостоятельной работы студента  $t^{\text{CPC}}$  и времени контактной работы  $t^{\text{Конт}}$  с одновременным уменьшением количества обучающихся  $count_s$ , задействованных в контактной работе и путем применения базы знаний  $Kb$  для минимизации влияния случайных и неслучайных деструктивных факторов при невозможности алгоритмического предсказания возникновения этих факторов в учебном процессе [65].

В параграфе 2.2 деструктивные факторы были разделены на два подмножества:  $\Phi_s$  – деструктивные факторы, влияющие на обучающегося и  $\Phi_l$  – деструктивные факторы, влияющие на преподавателя.

Для наполнения этих множеств обратимся к итогам проведенного в главе 1 анализа, в рамках которого были приведены положительно влияющие на результаты обучения факторы, которые были выделены в исследованиях Дж. Хэтти [104; 105].

Чтобы учесть в частной модели описанные в параграфе 1.1. факторы как деструктивные, их необходимо инвертировать. Например, для фактора  $\varphi_{15}$  «Повышение квалификации» инверсией описания будет «Преподаватель не повышает квалификацию» то есть повышение преподавателем своей квалификации будет положительным фактором, а ситуация, когда преподаватель не повышает квалификацию деструктивным фактором.

С учетом введенной инверсии факторов можно привести следующий состав множеств  $\Phi_s$  и  $\Phi_l$  в виде множеств (42) и (43)

$$\Phi_s = \{\varphi_1, \varphi_3, \varphi_4, \varphi_8, \varphi_9, \varphi_{10}, \varphi_{11}, \varphi_{12}, \varphi_{13}, \varphi_{14}\}, \quad (42)$$

$$\Phi_l = \{\varphi_3, \varphi_4, \varphi_5, \varphi_9, \varphi_{15}\}. \quad (43)$$

В двухуровневой модели взаимодействия участников учебного процесса зависимость результата обучающегося на  $i$ -ом этапе ИОТ от  $\Phi_s$  и  $\Phi_l$  описана с функций индивидуального уровня  $rs(tr_i, a^s, \Phi_s)$  и  $rl(tr_i, a^l, \Phi_l)$ . Модель взаимодействия участников учебного процесса описывает время на  $i$ -ом этапе ИУП как кортеж  $t_e^{\text{регл}} = \langle t_e^{\text{Конт}}, t_e^{\text{CPC}}, t_e^{\Delta}, t_e^{\text{CPП}} \rangle$ , а время на  $i$ -ом этапе ИОТ как кортеж  $t^{\text{Факт}} = \langle t_{\text{Факт}}^{\text{Конт}}, t_{\text{Факт}}^{\text{CPC}}, t_{\text{Факт}}^{\Delta}, t_{\text{Факт}}^{\text{CPП}} \rangle$ . В связи этим справедливы следующие уравнения (44)

$$\begin{aligned} t_e^{\text{регл}} &= t_e^{\text{Конт}} + t_e^{\text{CPC}} + t_e^{\Delta} + t_e^{\text{CPП}}, \\ t^{\text{Факт}} &= t_{\text{Факт}}^{\text{Конт}} + t_{\text{Факт}}^{\text{CPC}} + t_{\text{Факт}}^{\Delta} + t_{\text{Факт}}^{\text{CPП}}. \end{aligned} \quad (44)$$

Данные равенства демонстрируют, что при прохождении  $i$ -го этапа ИОТ время реагирования на деструктивные факторы  $t^{\Delta}$  влияет на общее распределение времени. Если  $t_{\text{Факт}}^{\Delta}$  превысит  $t_e^{\Delta}$ , то для сохранения соблюдения сроков освоения  $t^{\text{Факт}} \leq t_e^{\text{регл}}$  необходимо будет уменьшать  $t_{\text{Факт}}^{\text{Конт}}, t_{\text{Факт}}^{\text{CPC}}, t_{\text{Факт}}^{\text{CPП}}$  что негативно скажется на результате текущего этапа  $rtr(tr_i, A)$ . При применении НГО-агентов время контактной работы  $t_{\text{Факт}}^{\text{Конт}}$  может быть детализировано следующей формулой (45)

$$t_{\text{Факт}}^{\text{Конт}} = t_{\text{Факт}}^{\text{КонтПреп}} + t_{\text{Факт}}^{\text{КонтНГО}}, \quad (45)$$

где  $t_{\text{Факт}}^{\text{КонтПреп}}$  – время контактной работы обучающегося с преподавателем;  $t_{\text{Факт}}^{\text{КонтНГО}}$  – время контактной работы обучающегося с НГО-агентами.

Таким образом необходимо формализовать распределение времени участников учебного процесса для описания минимизации  $t_{\text{Факт}}^D$  и источников увеличения  $t_{\text{Факт}}^D$  при соблюдении сроков этапа и допустимого балла  $r_{ij}$  с учетом влияния деструктивных факторов.

Для этого введем коэффициенты  $w_{bs}$ ,  $w_{as}$ ,  $w_{bl}$ ,  $w_{al}$ , определяющие влияние на время через их связь с факторами  $\Phi_s$  и  $\Phi_l$ :

$w_{bs}$  – коэффициент влияния ошибок обучающегося на деятельность обучающегося;

$w_{as}$  – коэффициент влияния ошибок оценки на деятельность обучающегося;

$w_{bl}$  – коэффициент влияния ошибок обучающегося на деятельность преподавателя;

$w_{al}$  – коэффициент влияния ошибок оценки на деятельность преподавателя.

Для вычисления значений коэффициент  $w_{bs}$ ,  $w_{as}$ ,  $w_{bl}$ ,  $w_{al}$  необходимо перейти от множеств деструктивных факторов ко множествам величин Коэна для каждой группы деструктивных факторов, которые могут быть описаны формулами (46) и (47)

$$D_{\Phi_s} = \{d_1, d_3, d_4, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{14}\}, \quad (46)$$

$$D_{\Phi_l} = \{d_3, d_4, d_5, d_9, d_{15}\}, \quad (47)$$

где  $D_{\Phi_s}$  – множество величин Коэна для факторов  $\Phi_s$ ;

$D_{\Phi_l}$  – множество величин Коэна для факторов  $\Phi_l$

В таблице 3 представлены деструктивные факторы  $\varphi_i \in \Phi_s \cap \Phi_l$ , величины Коэна  $d_i$ , частота вхождения  $p_i$  и веса  $\omega_i$  деструктивных факторов, распределенные по связи деструктивных факторов с весами  $w_{bs}$ ,  $w_{as}$ ,  $w_{bl}$ ,  $w_{al}$ . Такое распределение позволяет учесть влияние деструктивных факторов на действия участников учебного процесса при прохождении ИУП.

Корреляция коэффициентов двухуровневой модели и некоторых значимых факторов по Хэтти приведена в таблице 3.

Таблица 3 – Связь коэффициентов двухуровневой модели и деструктивных факторов

Деструктивный фактор	Величина Коэна d	Частота вхождения $p_i$	Коэффициенты зависимости			
			Группа деструктивных факторов «Ошибки обучающихся»		Группа деструктивных факторов «Ошибки оценки»	
			Вес $\omega_i$ в коэффициенте $w_{bs}$ (проценты)	Вес $\omega_i$ в коэффициенте $w_{bl}$ (проценты)	Вес $\omega_i$ в коэффициенте $w_{al}$ (проценты)	Вес $\omega_i$ в коэффициенте $w_{as}$ (проценты)
$\varphi_1$	1,44	1	4,35	0,00	0,00	0,00
$\varphi_3$	0,9	2	0,00	8,70	0,00	8,70
$\varphi_4$	0,88	2	8,70	0,00	8,70	0,00
$\varphi_5$	0,88	1	0,00	0,00	4,35	0,00
$\varphi_8$	0,75	1	0,00	0,00	0,00	4,35
$\varphi_9$	0,74	4	17,39	17,39	17,39	17,39
$\varphi_{10}$	0,73	4	17,39	17,39	0,00	0,00
$\varphi_{11}$	0,72	1	4,35	0,00	0,00	0,00
$\varphi_{12}$	0,71	1	4,35	0,00	0,00	0,00
$\varphi_{13}$	0,67	2	8,70	8,70	0,00	0,00
$\varphi_{14}$	0,64	2	8,70	8,70	0,00	0,00
$\varphi_{15}$	0,62	2	0,00	0,00	8,70	8,70

Источник: составлено автором.

На основе данных таблицы 3 можно разделить деструктивные факторы на множества (48), (49), (50) и (51)

$$\Phi_s^{bs} = \{\varphi_1, \varphi_4, \varphi_9, \varphi_{10}, \varphi_{11}, \varphi_{12}, \varphi_{13}, \varphi_{14}\}, \quad (48)$$

$$\Phi_s^{as} = \{\varphi_3, \varphi_9, \varphi_{14}\}, \quad (49)$$

$$\Phi_l^{bl} = \{\varphi_4, \varphi_5, \varphi_9, \varphi_{15}\}, \quad (50)$$

$$\Phi_l^{al} = \{\varphi_3, \varphi_8, \varphi_{15}\}, \quad (51)$$

где  $\Phi_s^{bs}$  – деструктивные факторы использования времени обучающимся;  $\Phi_s^{as}$  – деструктивные факторы частоты возникновения ошибок обучающегося;  $\Phi_l^{bl}$  – деструктивные факторы использования времени преподавателем;  $\Phi_l^{al}$  – деструктивные факторы частоты возникновения ошибок оценки.

С учетом такого разделения каждое приведенное в таблице 3 значение веса  $\omega_i$  может быть рассчитано по следующей формуле (52)

$$\omega_i = \frac{p_i}{\sum p_i}, \quad (52)$$

где  $p_i$  – функция расчета частоты вхождения деструктивного фактора  $\varphi_i$  во множества деструктивных факторов  $\Phi_s^{bs}, \Phi_s^{as}, \Phi_l^{bl}, \Phi_l^{al}$  выражаемая формулой (53)

$$p_i = cnt(\varphi_i, \Phi_s^{bs}) + cnt(\varphi_i, \Phi_s^{as}) + cnt(\varphi_i, \Phi_l^{bl}) + cnt(\varphi_i, \Phi_l^{al}), \quad (53)$$

где  $cnt$  – функция подсчета количества элементов в множестве.

Для расчета коэффициентов  $w_{bs}, w_{as}, w_{bl}, w_{al}$  построим множества весов деструктивных факторов и множества величин Коэна, соответствующие множества  $\Phi_s^{bs}, \Phi_s^{as}, \Phi_l^{bl}, \Phi_l^{al}$  (54), (55), (56), (57), (58), (59), (60) и (61)

$$\Omega_s^{bs} = \{\omega_1, \omega_4, \omega_9, \omega_{10}, \omega_{11}, \omega_{12}, \omega_{13}, \omega_{14}\}, \quad (54)$$

$$\Omega_s^{as} = \{\omega_3, \omega_9, \omega_{14}\}, \quad (55)$$

$$\Omega_l^{bl} = \{\omega_4, \omega_5, \omega_9, \omega_{15}\}, \quad (56)$$

$$\Omega_l^{al} = \{\omega_3, \omega_8, \omega_{15}\}, \quad (57)$$

$$D_s^{bs} = \{d_1, d_4, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{14}\}, \quad (58)$$

$$D_s^{as} = \{d_3, d_9, d_{14}\}, \quad (59)$$

$$D_l^{bl} = \{d_4, d_5, d_9, d_{15}\}, \quad (60)$$

$$D_l^{al} = \{d_3, d_8, d_{15}\}. \quad (61)$$

С учетом этих множеств коэффициенты  $w_{bs}, w_{as}, w_{bl}, w_{al}$ , обеспечивающие описание зависимости фактического времени прохождения этапа ИУП от влияния деструктивных факторов для выполнения

требования по учету распределения времени участников учебного процесса, могут быть рассчитаны по следующим формулам (62)

$$\begin{aligned} w_{bs} &= \sum \omega_i^{bs} d_i^{bs}, \\ w_{as} &= \sum \omega_i^{as} d_i^{as}, \\ w_{bl} &= \sum \omega_i^{bl} d_i^{bl}, \\ w_{al} &= \sum \omega_i^{al} d_i^{al}. \end{aligned} \tag{62}$$

Таким образом можно сделать вывод, что распределение времени  $t^{\text{CPC}}, t^{\text{Конт}}, t^{\Delta}, t^{\text{СРП}}$  при взаимодействии участников учебного процесса зависит от двух групп деструктивных факторов – «Ошибки обучающихся» и «Ошибки оценки». Модель  $M^A$  предусматривает взаимодействие в рамках ИОТ  $Tr$ , которая состоит из этапов  $tr_i$ , то влияние деструктивных факторов на каждом этапе различно и зависит в том числе от текущего состояния преподавателя и обучающегося, что позволяет использовать модель  $M^A$  для синтеза методики оперативной корректировки ИОТ.

С учетом вычисления коэффициентов  $w_{bs}, w_{as}, w_{bl}, w_{al}$  можно привести модифицированное соотношение распределения времени с учетом деструктивных факторов, описываемой формулой (63)

$$t^{\text{Факт}} = w_{bs} w_{bl} t^{\text{Конт}} + w_{bs} t^{\text{CPC}}_{\text{Факт}} + w_{al} w_{as} t^{\Delta}_{\text{Факт}} + w_{al} t^{\text{СРП}}_{\text{Факт}}. \tag{63}$$

На каждом этапе ИОТ  $tr_i$  модель взаимодействия участников учебного процесса  $A^d$ , в котором может быть более одного агента-преподавателя. В таком случае, распределение времени можно модифицировать и выразить формулой (64)

$$t^{\text{Факт}} = \sum_{i=1}^{|A^L|} w_{bs} w_{bl} t^{\text{Конт}}_{\text{Факт}} + w_{bs} t^{\text{CPC}}_{\text{Факт}} + \sum_{i=1}^{|A^L|} w_{al} w_{as} t^{\Delta}_{\text{Факт}} + \sum_{i=1}^{|A^L|} w_{al} t^{\text{СРП}}_{\text{Факт}}. \tag{64}$$

Такая модификация учитывает тот факт, что модель взаимодействия участников учебного процесса описывает использование вспомогательных

НГО-агентов. Интеллектуализация каждого такого агента происходит за счет наличия у него контекста, формируемого на основе данных текущего взаимодействия и базы знаний.

Таким образом представленная частная модель позволяет учесть деструктивные факторы при фактическом распределении времени на каждом этапе, что обуславливает возможность более точного формирования ИОТ за счет корректировки временных параметров.

**Частная модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.** Частная модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой предназначена для формирования зависимости количества ошибок, а следовательно уровня ЗУН, от знаний интеллектуальных агентов и позволяет выполнить приведенное в параграфе 2.1 требование по минимизации ошибок обучающегося и ошибок оценки в условиях ограниченного времени самостоятельной и контактной работы и необходимости достижения заданного уровня освоения курса в установленные сроки. Выполнение данного требования достигается путем повышения точности и достоверности ответов агентов-преподавателей путем использования базы знаний  $Kb$  для формирования контекста  $ctx$  с применением алгоритма генерации с дополненной выборкой предназначена для минимизация влияния деструктивных факторов на НГО-агентов и минимизации ошибок во время контактной работы. С учетом распространенной проблемы галлюцинаций НГО время  $t_{\text{факт}}^{\Delta}$  при использовании НГО-агентов может как уменьшаться, так и увеличиваться.

Нейронные сети глубокого обучения, используемые в интеллектуальных агентах, могут быть созданы различными путями:

- 1) обучены с нуля;
- 2) получены в результате дообучения уже существующих НГО;

3) получение в результате применения алгоритма генерации с дополненной выборкой к существующей НГО.

Два первых метода трудозатратны и могут быть рассмотрены в рамках дальнейших исследований, лежащих за пределами настоящей диссертации. Существуют исследования [30; 34; 40; 135], которые показывают, что при использовании готовых НГО, когда обучение или дообучение невозможны, качество генерации НГО зависит от того, как сформирован контекст. В настоящее время выделяют два основных направления в реализации алгоритма генерации с дополненной выборкой:

- 1) классическая генерация с дополненной выборкой;
- 2) генерация с дополненной выборкой на основе знаний.

В рамках классической генерации с дополненной выборкой контекст формируется из слабо структурированной текстовой информации. При генерации с дополненной выборкой на основе знаний в качестве источника информации используются структурированные источники знаний: базы или графы знаний, что позволяет повысить качество выходных результатов НГО.

В связи с этим для минимизации ошибок обучающегося и ошибок оценки необходимо формализовать построение контекста интеллектуальных агентов, используемых в модели взаимодействия участников учебного процесса.

Средой взаимодействия участников учебного процесса является ИЭОС  $B$ , а возможные состояния среды для каждого курса с описываются графом ИУП  $Pl$ . Формирование ИОТ  $Tr$  заключается в построении цепи на графе  $Pl$ , при котором участники учебного процесса, в том числе интеллектуальные агенты, целенаправленно взаимодействуют и изменяют свои состояния.

Под состоянием НГО-агентов в настоящей работе понимается его текущий контекст  $ctx$ , который является совокупностью ситуационных данных, предоставляемых агенту в рамках входного запроса, которые необходимы для точной интерпретации намерений пользователя, генерации

релевантных ответов и поддержания связности диалога. Контекст позволяет наделить агента знаниями и формируется с помощью метода генерации с дополненной выборкой и необходим для задания агенту роли.

В рамках настоящей работы контекст интеллектуального агента формируется из:

- знаний, полученных из базы знаний  $Kb$ ;
- полученных восприятий  $percept$ .

С учетом этого необходимо детализировать описание контекста  $ctx$ , приведенное в модели взаимодействия участников учебного процесса. Контекст  $ctx$  можно представить следующей формулой (65)

$$ctx = \langle D', ctx^{pers}, ctx^{comp} \rangle, \quad (65)$$

где  $D'$  – проекция множества диалогов  $D$ , описанного в модели взаимодействия участников учебного процесса. Проекция старится таким образом, что в  $D'$  попадают только диалоги, в которых участвует данный интеллектуальный агент.

$ctx^{pers}$ , – персональный контекст, который описывает субъективные характеристики агента и может быть задан путем внесения текстовых утверждений об агенте, например «Ты – опытный программист», «Ты строгий, но справедливый».

Таким образом  $ctx^{pers}$  описывается как множество строк, содержащих утверждения в виде формулы (66)

$$ctx^{pers} = \{cp_1, cp_2, \dots, cp_m\}, \quad (66)$$

где  $ctx^{comp}$  – контекст компетенций, который описывает профессиональные компетенции агента. Контекст создается путем преобразования информации из баз знаний  $Kb^A$  в текстовые утверждения.

Таким образом  $ctx^{comp}$  описывается следующей формулой (67)

$$ctx^{comp} = \{cm_1, cm_2, \dots, cm_m\} \cup ctx^{crs}. \quad (67)$$

Контекст компетенций  $ctx^{comp}$  состоит из собственных утверждений, дополненных контекстом курса  $ctx^{crs}$ . Контекст курса  $ctx^{crs}$  формируется из текстовой или графической информации  $data$  каждого учебно-методического материала курса. Например, для курса «Структуры и алгоритмы обработки данных» по теме  $q_3$  «Типовые структуры описания абстрактных данных». Массив, список, стек, очередь будут использованы секции  $data$  следующих учебно-методических материалов  $e_i$ :

- 1)  $e_1(q_3)$  – текст лекции «Типовые структуры описания абстрактных данных. Массив, список, стек, очередь»;
- 2)  $e_2(q_3)$  – текст «Краткий методический материал для выполнения лабораторной работы «Массивы».

Контекст позволяет наделить агента знаниями, что позволяет снизить функции использования времени преподавателя  $\beta_l$  и частоты возникновения ошибок оценки  $\alpha_l$ . Контекст формируется с помощью метода генерации с дополненной выборкой и необходим для задания агенту роли. Задание ролей является одним из способов повышения качества работы основанного на НГО агента. Эффекты от задания ролей описаны в некоторых работах, посвященных обучению и многоагентным системам [31; 33; 36; 37; 41].

Построение контекста  $ctx^{comp}$  производится на основе базы знаний Kb и описывается моделью  $M^B$  (68)

$$< S, G, Kb, F >, \quad (68)$$

где  $S$  – модель компетенций, соответствующая курсу  $c$ ;

$G$  – граф знаний, дополняемый из модели компетенций  $S$ . Граф знаний может быть описан как совокупность вершин  $V$  и ребер  $G$  (69)

$$G = (V, E), \quad (69)$$

где вершина  $V$  является экземпляром одного из следующих классов: «Аксиома», «Концепт», «Синоним», «Термин», а ребра  $E$  задают отношения и иерархию между классами. Данная структура графа знаний обусловлена применением подхода слоеный пирог обучения онтологий» [85; 91].

В настоящей работе в графе знаний описываются интенсиональные знания, а утверждения об экземплярах вершин  $V$  графа знаний, то есть экстенсиональные знания содержатся в базе знаний  $Kb$ , которая может быть описана формулой (70)

$$Kb = Kb^A \cup Kb^O \cup R^k, \quad (70)$$

где  $Kb^A$  – компетентностный сегмент базы знаний, содержащий знания в сфере обучения программированию. Компетентностный сегмент базы знаний формируется на основе учебно-методических материалов  $E_c$  курса с;

$Kb^O$  – сегмент базы знаний общего назначения. В качестве сегмента базы знаний общего назначения использована открытая база знаний DBpedia;

$R^k$  – множество отношений, установленных между элементами  $Kb^A, G$  и  $Kb^O$ . Для описания связей понятий и концептов в основном применены отношения, используемые в OWL [118] и DBpedia, в частности модель данных Simple Knowledge Organization System [128] и Dublin Core [100].

$F$  – алгоритм формирования  $R^k$  путем связи элементов баз знаний  $kb_i^B \in Kb^B$  с вершинами графа  $G$ .

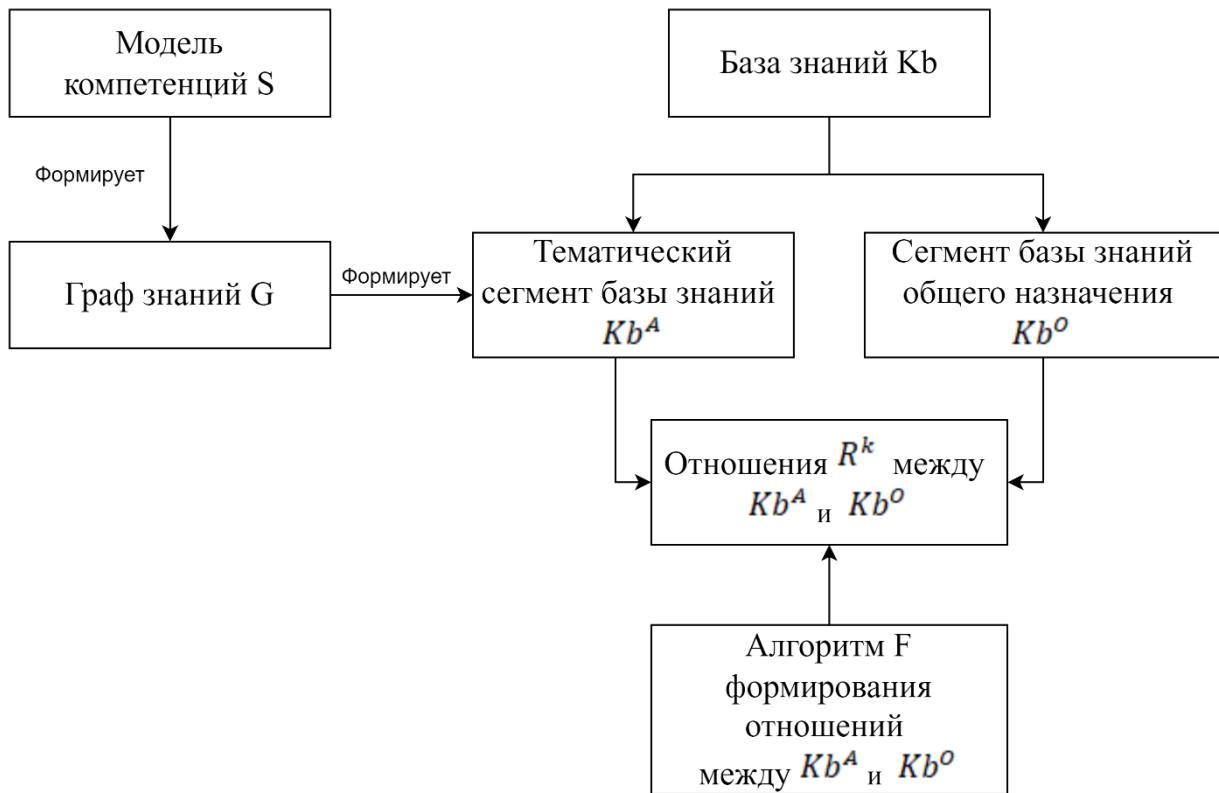
Такой подход к построению контекста позволяет снизить вероятность так называемых галлюцинаций НГО, поскольку контекст формируется из

проверенных источников знаний, сформированных на основе базы знаний общего назначения  $Kb^0$  и множества учебно-методических материалов  $E_c$ . Информация в данных источниках вносится экспертами, что обуславливает возможность ее использования. Необходимо заметить, что в настоящей работе в качестве области применения моделей выбрано обучение программированию, поскольку наличие в каждом учебно-методическом материале  $e_i$  множеств задач  $TK$  и контрольных примеров  $TS$  позволяет агенту-верификатору  $a_i^{JB}$  проверить достоверность генерируемого НГО текста и при необходимости скорректировать его до момента взаимодействия с обучающимся.

Структурная схема частной модели построения контекста представлена на рисунке 13.

Построение контекста  $ctx$  интеллектуального агента направлено на то, чтобы агент мог выполнять следующие действия:

- при построении ИОТ с применением МППР вычислять описанную в разделе 2.3 функцию  $ctrl(r_s(tr_i, a^s), \sum rl(tr_i, a^l))$  для принятия решения о переходе к следующему этапу ИУП;
- при прохождении этапа ИОТ повышать качество генерации ответов НГО для снижения влияния деструктивных факторов при распределении времени участников процесса  $t_{\text{Факт}}^{\text{Конт}}, t_{\text{Факт}}^{\Delta}, t_{\text{Факт}}^{\text{СРП}}$ , описанном в параграфе 2.4.



Источник: составлено автором.

Рисунок 13 – Структурная схема частной модели построения контекста

Таким образом, формализация построения ИОТ показала, что достижение заданного уровня освоения ЗУН за регламентированное время возможно при формализации распределения времени, позволяющей выделить виды взаимодействия, которое интеллектуальные агенты могут взять на себя и высвободить время преподавателя. Это обуславливает необходимость обеспечения качества работы НГО модели, что обеспечивается путем формирования контекста. Таким образом совокупность моделей позволяет описать достижение цели работы и формализовать критерии для оценки ее достижения.

## Выводы по второй главе

Во второй главе разработана модель взаимодействия участников учебного процесса для формирования индивидуальной образовательной

траектории при обучении программированию в интеллектуальной электронной образовательной системе, отличающаяся от известных моделей, предложенных А. А. Василенко [19], И. А. Кречетовым [13], Е. Ю. Благовым, И. А. Лещевой, С. А. Щербан [53], Д. А. Ульяновым [25], О. Х. З. Аль-Дулаими [10], F. Osborne и E. Motta [117], D. Edge [35] тем, что:

- двухуровневая зависимость индивидуальной образовательной траектории от текущих оценок показателей восприятия и реакции обучаемого с учетом действующих деструктивных факторов сформирована по результатам представления участников учебного процесса имитационными моделями с функциями интеллектуальных агентов, контекст которых формируется из базы знаний, содержащей онтологический сегмент с сущностями и отношениями, используемыми в семантической модели заданной предметной области;
- параметры ИОТ, формализуемой марковским процессом принятия решений, учитывают результаты прогноза показателей восприятия и реакции обучаемых на корректировку индивидуального учебного плана по данным сформированной нейросети глубокого обучения;
- учет данных о текущем уровне знаний, умений и навыков при прохождении учебной программы для своевременной корректировки следующего этапа ИОТ осуществляется по результатам выполняемой интеллектуальными агентами оценки параметров траектории.

Это позволяет сформировать зависимость достижения обучающимся необходимого для получения заданных компетенций уровня ЗУН в установленные образовательным стандартом сроки и на ее основе оперативно корректировать ИОТ для определения рациональной этапности обучения и необходимого времени.

## Глава 3

### Методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию

Представленная во второй главе модель взаимодействия участников учебного процесса позволила определить основные операции методики оперативного формирования ИОТ при обучении программированию с учетом регламентированного времени, которые участники учебного процесса могут потратить на заданный курс.

В силу того, что оперативное формирование ИОТ относится к задачам последовательного принятия решений в ситуациях, где результаты частично случайны и частично зависят от лица, принимающего решения, то указанный процесс характеризуется следующими особенностями:

- большое количество исходных данных;
- нечеткость представления исходной информации;
- значительное множество возможных решений;
- сложность математического представления целевой функции, определяющей заданный уровень ЗУН;
- необходимость оперативной корректировки образовательной траектории в зависимости от изменяющихся условий и исходных данных.

Для решения подобных задач в настоящее время эффективного полиномиального алгоритма не существует, в силу этого при решении задачи оперативного формирования ИОТ как правило применяется несколько алгоритмов, в том числе группа алгоритмов для больших размерностей исходных данных, и приближенные, которые работают достаточно быстро, но не гарантируют оптимального решения задачи.

В диссертации была разработана оригинальная методика оперативного формирования индивидуальной образовательной траектории, в которой

следующий этап определяется по результатам работы НГО-агентов. В методике на основании зависимостей, полученных на модели взаимодействия участников учебного процесса, предложены действия, позволяющие добиться приближенного решения задачи построения оптимальной ИОТ.

Эффективность методики в значительной мере определяется исходными данными об участниках учебного процесса и особенностями курса, который предполагается к прохождению.

### **3.1 Описание основных действий методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию**

Методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию и направлена на формирование оптимальной ИОТ, которая соответствует введенному в параграфе 1.4 критерию описывается формулой (71)

$$(R < R^{\min} \wedge T^{\max} - T \leq \varepsilon) \vee (R^{\min} < R \leq R^{\max} \wedge T^{\max} - T > \varepsilon), \quad (71)$$

где  $R$  – фактическая оценка уровня ЗУН в баллах при электронном обучении программированию;

$R^{\min}$  – удовлетворительная оценка уровня ЗУН в баллах согласно образовательной программе курса;

$R^{\max}$  – максимальная оценка уровня ЗУН в баллах согласно образовательной программе курса;

$T$  – фактическое время (в часах) изучения курса;

$T^{\max}$  – максимально допустимое время изучения курса (в часах), включающее регламентированное время изучения курса согласно образовательной программе дисциплины и время, затраченное преподавателем на работу по курсу;

$T^{min}$  – минимальное время (в часах) изучения курса, может быть оценено как время контактной работы  $t^{Kонт}$ ;

$\varepsilon$  – порог риска, то есть количество часов, после которого достижение заданного уровня ЗУН за регламентированное время считается невозможным.

Соблюдение критерия достигается при выборе оптимальной стратегии, которая была formalизована в параграфе 2.3 и заключается в следующем правиле: если разность фактического времени прохождения и регламентированного времени близка к порогу риска  $\varepsilon$ , то участники учебного процесса стремятся к достижению обучающимся минимально удовлетворительного уровня ЗУН  $R^{min}$ , в противном случае они стремятся к достижению максимального уровня ЗУН  $R^{max}$ .

Оптимальная стратегия ИЭОС задается зависимостями определяемыми на модели взаимодействия участников учебного процесса, тем самым обеспечивая исходные данные для разработки методики формирования ИОТ задаются на основе комплексирования параметров ИЭОС  $B = \langle A^S, A^L, C \rangle$  и базовых компонентов модели взаимодействия участников учебного процесса  $M^A = \langle A, c, Pl, Tr, Kb, \Phi \rangle$ . При этом частные модели для учета текущих значений параметров прохождения ИУП применяются на каждом его этапе. Такой учет позволяет оптимизировать время достижения заданного уровня ЗУН, что позволяет formalизовать методику оперативного формирования ИОТ программированию можно как функцию (72)

$$mtd(A, c, Pl, Kb, \Phi) \rightarrow Tr|optimum(B). \quad (72)$$

Как было сказано в параграфе 2.2, нахождение ИЭОС, в которой выполняется данный критерий, может быть выполнено путем применения нейронных сетей глубокого обучения. Кроме того, представление учебного процесса как системы «Преподаватель – Обучающийся» позволяет определить функции индивидуального и группового уровня, позволяющие formalизовать

достижение заданного уровня ЗУН за регламентированное время, то есть выполнить критерий оптимальности.

Указанная формализация определяет зависимости, обеспечивающие формирование методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию, основные действия которой представлены на рисунке 14.



Источник: составлено автором.  
Рисунок 14 – Основные действия методики

Согласно частной модели формирования ИОТ, представленной в параграфе 2.3, методика *mtd* может быть представлена как марковский процесс принятия решений, когда для расчета ценности следующего этапа ИОТ с учетом непройденных этапов ИУП используется численный метод решения уравнения Беллмана. Основными действиями этой методики являются:

- 1) формирование параметров ИУП;
- 2) формирование ИОТ;
- 3) оценка текущих параметров ИОТ;
- 4) корректировка ИОТ;

5) завершение.

Опишем основные действия методики с учетом адаптации частных моделей, описанных в параграфе 2.3.

### **3.2 Формирование индивидуальной образовательной траектории для применения в методике**

Формирование ИОТ в методике оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию необходимо описать, применение частной модели, а также базовых компонентов модели взаимодействия участников учебного процесса в основных действиях методики *mtd*.

**«Формирование параметров ИУП».** Осуществляется наполнение множества  $A$ . В параграфе 2.2 множество  $A$  было описано формулой (73)

$$A = \{a_1^L\} \cup \{a_1^S\} \cup A^J \cup \{a_0^M\} \cup A^{LL}, \quad (73)$$

где  $a_1^L \in A^L$  – преподаватель;

$a_1^S \in A^S$  – обучающийся;

$A^J = \{a^{rs}, a^{rl}, a^{ctrlle}, a^{test}\}$  – множество агентов-верификаторов, состоящее из  $a^{rs}$  – НГО-агента, используемого при вычислении функции индивидуального уровня  $rs(tr_i, a^s, \Phi_s)$ ;

$a^{rl}$  – НГО-агента, используемого при вычислении функции индивидуального уровня  $rl(tr_i, a^l, \Phi_l)$ ;

$a^{ctrlle}$  – НГО-агента, используемого при вычислении функции  $ctrlle(rs(tr_i, a^s, \Phi_s), \sum rl(tr_i, a^l, \Phi_l))$ ;

$a^{test}$  – агента, способного выполнять исходный код выполненных практических заданий;

$a_0^M$  – агент-модератор, обрабатывающий и маршрутизирующий все естественно-языковые сообщения, которыми обмениваются агенты.

Для маршрутизации сообщений между интеллектуальными агентами используется агент-модератор, который является подпрограммой, способной при обработке сообщений принимать следующие виды решений:

- 1) определение того, является ли сообщение, полученное от агента-обучающегося  $a_j^{HS}$ , допустимым, к недопустимым видам сообщений относятся сообщений с прямым запросом на решение задачи, предусмотренной курсом;
- 2) определение того, каким агентам отправлять сообщения агента-обучающегося  $a_j^{HS}$ , модератор может отправить сообщение одному из НГО-агентов  $a_i^L \in A^L$ , агенту верификатору  $a_i^J \in A^J$  или агенту-специалисту  $a_k^{HP} \in A^{HP}$ . При этом только модератор обладает информацией о том, что агент-специалист является человеком;
- 3) определение того, являются ли сообщения, полученные от агентов допустимыми для передачи обучающемуся  $a_j^{HS}$ , для определения допустимости модератор может перенаправить сообщение любому агенту кроме подмножества агентов  $A^{HT} \cup A^{HS} \cup A^M$ , таким образом организуется взаимодействие между агентами и совместная выработка ответов, при этом любой программный код передается модератором агенту-верификатору  $a_i^{JO}$ ;
- 4) определение того, какое итоговое сообщение будет отправлено обучающемуся  $a_j^{HS}$ , обучающемуся может быть отправлено как одно, так и несколько сообщений с указанием того, какими агентами были сгенерированы сообщения.

$A^{LL} = \{a^{ll}, a^{lp}, a^{la}\}$  – множество НГО-агентов взаимодействующих с обучающимся  $a_1^S$  и являющихся ассистентами преподавателя  $a_1^L$ , состоящее из  $a^{ll}$  – НГО-агента с ролью «Преподаватель»,  $a^{lp}$  – НГО-агента с ролью «Программист», НГО-агента с ролью «Аналитик». Необходимо заметить, что модель взаимодействия участников учебного процесса не задает конкретный

состав НГО-агентов, что позволяет определить его на этапе применения методики. Роли агентов задаются путем наполнения, описанного в параграфе 2.3 персонального контекста  $ctx^{pers}$ .

Для НГО-агентов  $A^{LL}$  и агентов-верификаторов  $a^{rs}, a^{rl}, a^{ctrl}$  необходимо выбрать используемую большую языковую модель  $mdl$ . При выборе модели необходимо провести анализ существующих моделей по следующим критериям:

- качество генерации с учетом использования генерации с дополнительной выборкой;
- соблюдение моделью ограничений, заданных входными параметрами и параметрами среды.

Выбор модели проводится с применением предложенного Т. Саати метода анализа иерархий [3].

Агент-верификатор  $a^{test}$  может быть реализован на базе систем класса Online Judge, в основе которых лежит способность запустить программный код с заданными входными данными и проверять результаты на соответствие заранее заданным проверочным значениям. Агент-верификатор обрабатывает оценочные средства  $k_i$  курса, таким образом, что элемент оценочного средства  $data$  выражается как множество контрольных примеров  $TS$ , каждый элемент  $test_i$  содержит входные данные  $in$ , выходные данные  $out$  и функцию  $f$ , которая описывает способ проверки соответствия  $in$  и  $out$  в виде формул (74), (75) и (76)

$$TS = \{ts_1, ts_2, \dots, ts_k\}, \quad (74)$$

$$ts_i = \{ \langle in, out, f \rangle_1, \langle in, out, f \rangle_2, \dots, \langle in, out, f \rangle_t \}, \quad (75)$$

$$f(in) = out. \quad (76)$$

Приведем пример из опыта курса «Структуры и алгоритмы обработки данных». В качестве примера дана простейшая задача на обработку массива: «Дан одномерный массив строк. Отсортируйте его методом пузырька».

Входными данными  $in$  является следующий массив: {Аaaaa, АААА, aaa, aaaaaaa}, проверочными выходными данными  $out$  является массив {aaaaaaaa, Aaaaa, АААА, aaa}, функцией проверки  $f$  следующий код на языке Java: `Arrays.equals(in, out)`. Оценочное средство  $k_i$  позволяет проверить правильность составления программы для сортировки массива на языке Java.

После определения состава агентов из множества курсов С должен быть выбран курс для прохождения с. Выбор курса с позволяет задать  $R^{\min}, R^{\max}, T^{\max}, T^{\min}$  необходимые для применения критерия оптимальности (77)

$$(R < R^{\min} \wedge T^{\max} - T \leq \varepsilon) \vee (R^{\min} < R \leq R^{\max} \wedge T^{\max} - T > \varepsilon). \quad (77)$$

Следующим входным параметром методики является ИУП  $Pl$ , который формируется следующим образом.

При составлении курса с вносятся все вершины  $V_{pl}$  графа  $Pl$  с указанием всех параметров вершины и ребра  $E_{pl}$ . При этом вероятность р перехода по ребру для каждого ребра  $e_{pl_k}$  рассчитывается исходя из истории прохождения обучающимся  $a_1^S$  курсов в ИЭОС.

Анализ предыдущих прохождений курса  $c_i$  другими обучающимися позволяет оценить сложность учебно-методических материалов  $e_j^C$  на основе их прохождения. Оценка сложности учебно-методических материалов необходима уточнения значений матрицы известных вероятностей перехода  $P(pl_j|pl_i, act)$ .

Справедливо отношение  $\sum_{pl_j \in Pl} P(pl_j|pl_i, act) = 1$ , обуславливающее то, что в общем случае вероятность перехода из вершины  $pl_i$  в инцидентные вершины  $pl_j$  и  $pl_k$  может полагаться одинаковой. Однако, практика показывает, что этапы повышенной сложности, как правило, изучаются реже, поэтому заданное по умолчанию равное распределение вероятностей необходимо скорректировать.

Ручная корректировка, проводимая экспертным путем, когда преподаватель самостоятельно при составлении ИУП задает вероятности перехода, трудозатратна, поэтому она может быть замена следующим подходом.

При первом прохождении курса  $c_i$  матрица переходных вероятностей  $P$  будет заполнена одинаковыми значениями вероятностей для альтернативных этапов ИУП. При последующих прохождениях курса матрица переходных вероятностей  $P$  должна заполняться на основе пройденных ИОТ.

Формирование ИУП с учетом профиля обучающегося и предыдущих изучений заданного курса с другими обучающимися может происходить путем корректировки вероятности  $p_{ij}$  для перехода из этапа добавления этапа  $tr_i$  этапа ИУП  $pl_i$  к этапу  $pl_j$ . При этом должно соблюдаться ограничения, описываемые формулами (78) и (79)

$$R^{min} \leq \sum R(pl_i, act, pl_j) \leq R^{max}, \quad (78)$$

$$T^{min} \leq \sum t_{pl_i}^{\text{регл}} \leq T^{max}, \quad (79)$$

где  $R(pl_i, act, pl_j) = rtr$  – предполагаемая оценка за этап ИУП;

$t_{pl_i}^{\text{регл}}$  – заданное время на прохождение  $i$ -го этапа ИУП.

Соблюдение данного ограничения соответствует цели работы в части обеспечения достижения требуемого уровня знаний, умений и навыков за установленное время.

База знаний  $Kb$  формируется из двух сегментов. Первым сегментом  $Kb^0$  является сегмент общего назначения, созданный на основе открытой базы знаний DBpedia. Вторым сегментом является компетентностный сегмент  $Kb^A$ , который формируется на основе учебно-методических материалов  $E$  курса с.

Множество деструктивных факторов  $\Phi$  инициализируется как объединение подмножеств  $\Phi_s$  и  $\Phi_l$ , которые состоят из деструктивных факторов, описываемых формулами (80) и (81)

$$\Phi_s = \{\varphi_1, \varphi_3, \varphi_4, \varphi_8, \varphi_9, \varphi_{10}, \varphi_{11}, \varphi_{12}, \varphi_{13}, \varphi_{14}\}, \quad (80)$$

$$\Phi_l = \{\varphi_3, \varphi_4, \varphi_5, \varphi_9, \varphi_{15}\}. \quad (81)$$

Описание факторов приведено в таблице 4.

Таблица 4 – Описание деструктивных факторов

№	Обозначение	Деструктивный фактор
1	$\varphi_1$	Ученики не оценивают свой прогресс и ошибки
2	$\varphi_3$	Регулярный контроль и предоставление результатов не проводятся
3	$\varphi_4$	Совместное выполнение заданий с небольшой группой учащихся и их последующий разбор не проводятся
4	$\varphi_5$	Способные ученики не осваивают учебную программу в опережающем режиме
5	$\varphi_8$	Материал не понятен обучающимся
6	$\varphi_9$	Обучающиеся не имеют возможности задавать вопросы, получаться разъяснения, обобщать и в целом вести диалог с преподавателем
7	$\varphi_{10}$	Преподаватель не дает конструктивные комментарии преподавателя, направленные на улучшение.
8	$\varphi_{11}$	Отношения «учитель-ученик» не складываются
9	$\varphi_{12}$	Индивидуальная и групповая работа не ведутся или не сбалансированы
10	$\varphi_{13}$	Предыдущий опыт и достижения обучающегося не учитываются
11	$\varphi_{14}$	Самоанализ не проводится
12	$\varphi_{15}$	Преподаватель не повышает квалификацию

Источник: составлено автором по материалам [104; 105].

Эти факторы определяют, как будет происходить распределение фактического времени прохождения этапа ИУП. Предложенная в параграфе 2.3 частная модель распределения времени позволяет снизить влияние этих деструктивных факторов путем уменьшения времени, необходимого на реагирование на них.

Роль действия «Формирование параметров ИУП» в рамках методики *mtd* крайне важна, поскольку позволяет использовать ИУП для задания начальных вероятностей перехода между этапами и таким образом formalизовать ИОТ как полумарковский случайный процесс с частичной моделью.

**«Формирование ИОТ».** Согласно описанной в параграфе 2.3 частной модели формирования ИОТ траектория  $Tr$  является путем на графе  $Pl$ , ребра которого в терминах марковского процесса принятия решений являются переходом вида  $(pl_i, act) \rightarrow pl_j$ , описывающим зависимость выбора следующего шага от текущего времени и уровня ЗУН. Переход по ребрам осуществляется с применением функции полезности  $R(pl_i, act, pl_j)$ , описываемой формулой (82)

$$R(pl_i, act, pl_j) = rtr(tr_i, A) + E[rtr(pl_j, A)] + fw(pl_j, \varepsilon), \quad (82)$$

где  $rtr(tr_i, A)$  – фактическая оценка, полученная за пройденный этап ИОТ  $tr_i$ ;

$E[rtr(pl_j, A)]$  – математическое ожидание оценки на следующем этапе ИУП  $pl_j$ , включаемом в ИОТ;

$fw(pl_j, \varepsilon)$  – функция корректировки ожидаемой полезности с учетом суммарного времени, затраченного на прохождение ИОТ и порога риска  $\varepsilon$  – количества часов, после которого достижение заданного уровня ЗУН за регламентированное время считается невозможны.

Добавление функции  $fw(pl_j, \varepsilon)$  в расчет функции полезности обусловлено необходимостью учета, введенного в п. 1.4 критерия оптимальности (83)

$$(R < R^{\min} \wedge T^{\max} - T \leq \varepsilon) \vee (R^{\min} < R \leq R^{\max} \wedge T^{\max} - T > \varepsilon). \quad (83)$$

Таким образом в основу методики оперативного формирования ИОТ положено последовательное применение функции  $R(pl_i, act, pl_j)$  на всех этапах прохождения курса.

Определение того, к какому этапу ИУП будет осуществлен переход, в формализме МППР зависит от описанного в частной модели формирования ИОТ коэффициента дисконтирования  $\gamma$ , вычисляемого по формуле (85)

$$\gamma = \begin{cases} 0,9, & \text{если } \sum R(pl_i, act, pl_j) < R^{min} \\ 0,1, & \text{если } \sum R(pl_i, act, pl_j) \geq R^{min} \end{cases}. \quad (85)$$

Действия агента описываются стратегией выбора  $\pi$ , заключающейся в максимизации ценности состояния и описываемой формулой (86)

$$V(\pi, tr_i) = \max^{act} \sum_{pl_j} P(pl_j | pl_i, act) [R(pl_i, act, pl_j) + \gamma V(\pi, pl_j)]. \quad (86)$$

При этом функция полезности  $R(pl_i, act, pl_j)$  зависит от функции группового уровня  $rtr(tr_i, A)$  и функции корректировки  $fw(pl_j, \varepsilon)$ .

Для определения порога риска воспользуемся формулой (87)

$$T^{max} - T^{ост} = T^{max} - \sum_{i+1}^N t_e^{\text{регл}}, t_e^{\text{регл}} \in Pl^*, \quad (87)$$

где  $Pl^*$  – непройденная часть ИУП.

Для определения  $\sum_{i+1}^N t_e^{\text{регл}}, t_e^{\text{регл}} \in Pl^*$  в применим алгоритм Левита, который позволяет найти кратчайшее расстояние между вершинами графа. Выразим  $T^{ост}$  через применение этого алгоритма к подграфу  $Pl^*$  следующей формулой (88)

$$T^{ост} = \min^{L\psi} (\sum_{i+1}^N t_e^{\text{регл}}), t_e^{\text{регл}} \in Pl^*. \quad (88)$$

Найденный при вычислении описанной в параграфе 2.3 функции  $fw(pl_j, \varepsilon)$  кратчайший путь является оптимальным по времени, но не учитывает достижение требуемого уровня освоения. Для расчета ценности этапа  $pl_j$  с учетом непройденных этапов ИУП может быть использован один

из численных методов решения уравнения Беллмана, например метод обратного прохода, описываемой формулой (89)

$$V(pl_j) = \sum_{act} \pi(act|pl_j) \sum_{pl_i} P(pl_i|pl_j, act) [R(pl_i, act, pl_j) + \gamma V(pl_i)]. \quad (89)$$

Таким образом, выбор описанной в параграфе 2.3 оптимальной стратегии заключается в следующем правиле: если разность фактического времени прохождения и регламентированного времени близка к порогу риска  $\varepsilon$ , то участники учебного процесса стремятся к достижению обучающимся минимально удовлетворительного уровня ЗУН  $R^{min}$ , в противном случае они стремятся к достижению максимального уровня ЗУН  $R^{max}$ .

**«Оценка текущих параметров ИОТ».** Как было сказано выше, оценка достижения заданного уровня освоения дисциплин заключается в соблюдении неравенства  $R^{min} \leq R \leq R^{max}$ .

Для этого на каждом этапе  $tr_i$  ИОТ с помощью функции *ctrlle* вычисляется уровень фактического освоения.

Алгоритм работы функции *ctrlle* состоит из следующих шагов:

1) Определяется, какое оценочное средство  $k_i$  было применено для заданного этапа  $tr_i$ .

2) Если было применено оценочное средство с типом  $type_1$ , то происходит переход к шагу 3, если было применено оценочное средство с типом  $type_2$ , то происходит переход к шагу 8, иначе происходит переход к шагу 12).

3) Агент-модератор  $a_0^M$  передает диалог с обучающимся агенту-верификатору  $a_i^{JO} \in A^J$  для формальной проверки ответов.

4) Агент-верификатор  $a_i^{JO}$  из множества  $TASK$  выбирает  $task_i$  с типом  $type_1$  для прохождения обучающимся теста  $test_i$ .

5) Обучающийся вводит ответы *input*.

6) Функция  $ctrl_e$  вычисляется как результат вычисления  $f(input) = output$  для теста  $test_i$  с возможными оценками  $\{1, \dots, N\}$ .

7) Агент-модератор  $a_0^M$  передает диалог с обучающимся агенту-преподавателю  $a_1^L$  для обсуждения результатов теста.

8) Агент-модератор  $a_0^M$  передает диалог с обучающимся агенту-верификатору, способному выполнить программный код и запустить тесты.

9) Агент-верификатор  $a^{test}$  из множества  $TASK$  выбирает  $task_j$  с типом  $type_2$  для прохождения обучающимся теста  $test_j$ .

10) Обучающийся вводит программный код и запускает тесты  $test_j$ , предложенные агентом-верификатором.

11) Функция  $ctrl_e$  вычисляется как результат вычисления  $f(input) = output$  для теста  $test_j$  с возможными оценками  $\{1, \dots, N\}$ , где оценка определяется на основе количества допущенных ошибок и итераций, затраченных на исправление.

12) Агент-модератор  $a_0^M$  передает диалог с обучающимся агенту-верификатору  $a^{rs}$  для вычисления оценки опроса.

13) Агент-верификатор  $a^{rs}$  вычисляет оценку.

Результат функции  $ctrl_e$  сравнивается со значениями заданного уровня освоения на этапе ИОТ  $tr_i : R_e^{min}$  в виде двойного неравенства (90)

$$R_e^{min} \leq ctrl_e \leq R_e^{max}. \quad (90)$$

При принятии решения о переходе к следующему этапу ИОТ или повторному прохождению текущего этапа ИОТ производится предварительное вычисление значения общего уровня освоения  $R$

Далее вычисляется следующее неравенство (91)

$$|\sum R_{ei}^{min} - rtr(tr_i, A)| \leq |\sum R_{ei}^{max} - rtr(tr_i, A)|, \quad (91)$$

где  $R_{ei}^{min}$  – минимальный уровень ЗУН на предыдущих  $i - 1$  этапах ИУП;  
 $R_{ei}^{max}$  – максимальный уровень ЗУН на предыдущих  $i - 1$  этапах ИУП.

Данное неравенство показывает, к верхней или нижней границе допустимых уровней ЗУН ближе фактический уровень ЗУН. В сочетании с информацией о затраченных на прохождение ИОТ времени это неравенство позволяет принять решение о том, переходить ли к следующему этапу ИОТ или требуется повторение текущего этапа. Однако, для принятия решения необходима оценка соблюдения установленных сроков.

Как было сказано выше, оценка соблюдения установленных сроков заключается в соблюдении неравенства  $T \leq T^{\text{регл}}$ .

Для этого на каждом этапе ИОТ  $tr_i$  производится подсчет времени (в часах) на прохождение обучающимся  $i$ -го этапа ИОТ  $T_i$  и  $T_i^l$  времени (в часах), затраченного преподавателем на прохождение обучающимся  $i$ -го этапа ИОТ. Подсчет времени осуществляется по формулам (92) и (93)

$$T_i = t_{\text{Факт}}^{\text{Конт}} + t_{\text{Факт}}^{\text{СРС}}, \quad (92)$$

$$T_i^l = t_{\text{Факт}}^{\Delta} + t_{\text{Факт}}^{\text{СРП}}, \quad (93)$$

где  $t_{\text{Факт}}^{\text{Конт}}$  – фактическое время контактной работы обучающегося и преподавателя;

$t_{\text{Факт}}^{\text{СРС}}$  – фактическое время самостоятельной работы обучающегося;

$t_{\text{Факт}}^{\Delta}$  – фактическое время, затрачиваемое на реагирование на деструктивные факторы;

$t_{\text{Факт}}^{\text{СРП}}$  – фактическое время самостоятельной работы преподавателя.

Таким образом на каждом этапе ИОТ  $tr_i$  производится вычисление следующего неравенства (94)

$$\sum T_i + \sum T_i^l \leq \sum T_{ei}^{\text{регл}}, \quad (94)$$

где  $T_{ei}^{\text{регл}}$  – установленные сроки освоения учебных модулей на предыдущих  $i - 1$  этапах ИУП.

На основании данного неравенства с учетом значения текущего уровня освоения  $R$  принимается решение о возможности перехода к следующему этапу ИОТ путем рассмотрения их совокупности. Индивидуальная образовательная траектория может быть скорректирована как таким образом, чтобы освоить минимально необходимый материал в случае риска нарушения сроков, так и для прохождения ИОТ с максимальным погружением в материал дисциплины, если динамика обучающегося позволяет это.

**«Корректировка ИОТ»** Корректировка ИОТ тесно связана с формированием ИОТ. Эти действия определяют то, как будет пройден текущий этап ИОТ и то, какой следующий этап ИУП должен быть выбран с учетом текущих параметров прохождения. Таким образом корректировка ИОТ может быть описана следующим алгоритмом:

- 1) Для прохождения  $d_j$  текущего этапа ИОТ  $tr_i$  на сообщении  $me_k$  агентом-модератором  $a_0^M$  определена необходимость в оценке текущего уровня освоения.
- 2) Агент-модератор  $a_0^M$  передает диалог одному из агентов-верификаторов  $a_i^J \in A^J$ . Агент-верификатор вычисляет текущий уровень ЗУН с помощью функции  $rtr(tr_i, A)$ .
- 3) Если текущий уровень ЗУН  $r_{dj} > R_e^{\min}$ , то выполняется переход к шагу оценки времени. В противном случае прохождение  $d_j$  продолжается.
- 4) Если текущее время прохождения  $t_{dj} < t_e^{\min}$ , то происходит переход к шагу построения углубленного сценария, в противном случае происходит переход к шагу построения выравнивающего сценария.
- 5) Углубленный сценарий формируется в условиях, когда  $r_{dj} > R_e^{\min} \wedge t_{dj} < t_e^{\min}$ , то есть конкретный студент легко справляется с

выполнением рабочей программы, поэтому на нем выбирается такой следующий этап траектории  $tr_{i+1}$ , в котором содержится базовый или углубленный уровень учебно-методических материалов  $e_j^C$ , что обуславливает формирование углубленного контекста агента-преподавателя  $a_r^{ll}$ .

6) Выравнивающий сценарий формируется условиях, когда  $r_{dj} \geq R_e^{min} \wedge t_{dj} \geq t_e^{min}$  то есть конкретный студент с трудом справляется с выполнением рабочей программы, поэтому на нем выбирается такой следующий этап траектории  $tr_{i+1}$ , в котором содержится базовый или упрощенный уровень учебно-методических материалов  $e_j^C$ , что обуславливает формирование упрощенного контекста агента-преподавателя  $a_r^{ll}$ .

7) Происходит переход к этапу ИОТ  $tr_{i+1}$ .

После прохождения курса анализируется отличие фактической ИОТ от ИУП как в части проверки параметров на этапах  $tr_i$ , так в части выбора альтернативных сценариев прохождения. На основании результатов анализа плановая ИОТ может быть скорректирована.

Созданная методика оперативного формирования ИОТ позволяет за счет использования описанных в модели взаимодействия участников учебного процессе интеллектуальных агентов определить момент достижения минимального уровня освоения и принять решение о следовании ранее запланированной ИОТ или ее изменении и выборе альтернативных модулей, что позволяет динамически корректировать сценарии прохождения ИОТ по результатам вычисления функции полезности.

### **3.3 Распределение времени участников учебного процесса**

Как было показано в параграфе 2.3 распределение времени участников учебного процесса зависит от влияния деструктивных факторов. В целях применения методики *mtd* деструктивные факторы можно связать с основными действиями методики. Такая связь отражена в таблице 5.

Таблица 5 – Связи деструктивных факторов с действиями методики

Деструктивные факторы	Формирование параметров ИУП	Оценка текущих параметров ИОТ	Формирование ИОТ и Корректировка ИОТ
$\Phi_s^{bs}, \Phi_s^{as}$	$\varphi_4$	$\varphi_{12}$	$\varphi_1, \varphi_4, \varphi_9, \varphi_{11}, \varphi_{12}, \varphi_{13}$
$\Phi_l^{bl}, \Phi_l^{al}$	$\varphi_5$	$\varphi_4$	$\varphi_5$

Источник: составлено автором.

Анализ таблицы 5 позволяет прийти к выводу о том, что подготовка интеллектуальных агентов зависит от деструктивных факторов, влияющих на компетенции и возможности контактной работы. Формирование и корректировка индивидуальной образовательной траектории имеют наибольшую зависимость от деструктивных факторов, допущенных ошибок и случайных эффектов, при этом большое влияние имеет контактная работа и оценка текущего уровня освоения.

Анализ результатов прохождения индивидуальной образовательной траектории в основном связан с рисками неправильного определения итогового уровня компетенций для дальнейшего использования. Таким образом пары «деструктивный фактор – действие методики» должны учитываться при использовании в методике некоторых базовых элементов модели взаимодействия участников учебного процесса. Результаты такого соотнесения приведены в таблице 6.

Таблица 6 – Связь базовых элементов модели взаимодействия участников учебного процесса с действиями методики и деструктивными факторами

Элемент модели взаимодействия участников учебного процесса	Группы деструктивных факторов	
	$\Phi_s^{bs}, \Phi_s^{as}$	$\Phi_l^{bl}, \Phi_l^{al}$
$A$ – множество участников учебного процесса	Оценка текущих параметров ИОТ - $\varphi_{12}$ Формирование ИОТ- $\varphi_{12}$ Корректировка ИОТ - $\varphi_{12}$	Оценка текущих параметров ИОТ - $\varphi_4$
$Pl$ – индивидуальный учебный план	Формирование параметров ИУП - $\varphi_4$	Формирование параметров ИУП - $\varphi_5$
$Tr$ – индивидуальная образовательная траектория	Формирование ИОТ $\varphi_1, \varphi_4, \varphi_9, \varphi_{11}, \varphi_{12}, \varphi_{13}$ Корректировка ИОТ $\varphi_1, \varphi_4, \varphi_9, \varphi_{11}, \varphi_{12}, \varphi_{13}$	Формирование ИОТ - $\varphi_5$ Корректировка ИОТ - $\varphi_5$

Источник: составлено автором.

Для адаптации агентов-участников учебного процесса при выборе состава агентов в условиях действия деструктивного фактора  $\varphi_{12}$  «Индивидуальное или групповое обучение» необходимо определить состав агентов на основе НГО из множества  $A^{LL}$ . Выше было показано, что использование нескольких НГО-агентов положительно влияет на качество ответов, в связи с этим в множестве  $A^{LL}$  предусмотрен агента-практик  $a_i^{LP}$ , который может имитировать совместное обучение и, вместе с агентом-преподавателем  $a_r^{ll}$  давать обратную связь, что положительно влияет на минимизацию фактора  $\varphi_{12}$ .

Адаптация модели  $M^A$  к фактору  $\varphi_4$  «Микрообучение» при выборе состава агентов, формировании основных и альтернативах сценариев прохождения ИОТ обуславливает наличие агента-преподавателя  $a_r^{ll}$ , который может выполнять задания и производить их разбор вместе с обучающимся в условиях, когда временные затраты на это ограничены временем самостоятельной работы студента  $t^{CPC}$  и могут быть дополнены частью времени контактной работы  $t^{Конт}$ . В то же время преподаватель-человек имеет может уделить каждому обучающемуся только время, описанное соотношением  $\frac{t^{Конт}}{count_s}$ , где  $count_s$  – количество студентов в группе обучающихся. В общем случае справедливо неравенство (95)

$$t^{CPC} \gg \frac{t^{Конт}}{count_s}. \quad (95)$$

В связи с этим можно сделать вывод, что применение нескольких НГО-агентов существенно минимизирует влияние фактора  $\varphi_{12}$ . Вместе с тем агенты должны применяться в обязательной связке с агентами-верификаторами, например, агентом  $a_i^{JO}$ , способным выполнить и проверить на соответствие тестовым примерам, программный код, предлагаемый другими агентами обучающемуся.

При формировании ИУП модель  $M^A$  с помощью возможности задания альтернативных сценариев прохождения позволяет снизить влияние деструктивного фактора  $\varphi_5$  «Ускорение в обучении» в связи с тем, что при корректировке ИОТ на каждом этапе оценивается уровень достижения заданного уровня освоения модулей учебных дисциплин и решение о выборе следующего этапа принимается исходя неравенства (96)

$$|\sum R_{ei}^{min} - rtr(tr_i, A)| \leq |\sum R_{ei}^{max} - rtr(tr_i, A)|, \quad (96)$$

где  $rtr(tr_i, A)$  – значение уровня освоения на текущем этапе ИОТ;

$R_{ei}^{min}$  – минимальный уровень ЗУН на предыдущих  $i - 1$  этапах ИУП;

$R_{ei}^{max}$  – максимальный уровень ЗУН на предыдущих  $i - 1$  этапах ИУП.

Вторым критерием выбора следующего этапа является соблюдение неравенства (97)

$$\sum T_i + \sum T_i^l \leq \sum T_{ei}^{\text{регл}}, \quad (97)$$

где  $T_{ei}^{\text{регл}}$  – установленные сроки освоения учебных модулей на предыдущих  $i - 1$  этапах эталонной ИОТ  $Tr^0$ .

Подсчет времени осуществляется по следующим формулам (98)

$$\begin{aligned} T_i &= t_{\text{Факт}}^{\text{Конт}} + t_{\text{Факт}}^{\text{СРС}}, \\ T_i^l &= t_{\text{Факт}}^{\Delta} + t_{\text{Факт}}^{\text{СРП}}, \end{aligned} \quad (98)$$

где  $t_{\text{Факт}}^{\text{Конт}}$  – фактическое время контактной работы обучающегося и преподавателя;

$t_{\text{Факт}}^{\text{СРС}}$  – фактическое время самостоятельной работы обучающегося;

$t_{\text{Факт}}^{\Delta}$  – фактическое время, затрачиваемое на реагирование на деструктивные факторы;

$t_{\text{Факт}}^{\text{СРП}}$  – фактическое время самостоятельной работы преподавателя.

Выбор альтернативных сценариев прохождения зависит не только от достижения минимального уровня, но и от того, за какое время он достигнут. Для обучающегося, быстро достигающего результатов, будет предложена такая траектория, которая позволит получить углубленные знания с соблюдением установленного срока.

Прохождение ИОТ  $Tr$  связано с несколькими деструктивными факторами  $\varphi_1$  «Самооценка знаний обучающимся»,  $\varphi_9$  «Взаимообучение»,  $\varphi_{11}$  «Отношения «учитель-ученик»»,  $\varphi_{13}$  «Предыдущие достижения».

В модели  $M^A$  влияние факторов  $\varphi_1, \varphi_{13}$  учитывается в функции оценки  $ctrl$  поскольку расчет текущего уровня освоения  $R$  производится с учетом суммы оценок за пройденные этапы  $tr_i$ .

Учет факторов  $\varphi_9$  и  $\varphi_{13}$  при прохождении ИОТ также связан с функцией оценки  $ctrl$ . Алгоритм функции предусматривает, что результат вычисления  $f(input) = output$  для теста  $test_i$  с возможными оценками  $\{1, \dots, N\}$  будет передан агентом-модератором  $a_0^M$  агенту-преподавателю  $a_r^l$  для обсуждения результатов задачи. Необходимо заметить, что снижение влияния этих деструктивных факторов также достигается за счет того, что время самостоятельной работы обучающегося  $t^{\text{СРС}}$  превращается в более эффективное время контактной-работы  $t^{\text{Конт}}$ .

Частная модель распределения времени участников учебного процесса адаптирована для применения в методике оперативного формирования ИОТ путем учета конкретных деструктивных факторов, влияющих на временные параметры прохождения этапа ИОТ, за счет построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

### 3.4 Построение контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой

В параграфе 2.3 предложена частная модель построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой представления знаний агентов. Она используется при выполнении действий «Формирование ИОТ», «Оценка текущих параметров ИОТ» и «Корректировка ИОТ» методики.

Для этого база знаний  $kb_i$  агента-преподавателя  $a_r^l$  должна содержать связи с вершинами графа знаний  $G$ , полученными с помощью функции  $convert^g(s_i)$  из модели компетенций  $s_i$ . Это позволяет интеллектуальному агенту определять, соответствует ли текущий уровень ЗУН программы уровнем, предусмотренным образовательным стандартом.

При выборе состава агентов и формировании их компетенций влияние деструктивного фактора  $\varphi_{15}$  «*Повышение квалификации*» учитывается с помощью применения к учебно-методическим материалам  $e_i$  функции  $to\_kb$ , которая устанавливает отношения между отношениями  $R^k$ , установленных между элементами  $Kb^A, G$  и  $Kb^O$ , где  $Kb^A$  – специализированная база знаний, содержащая знания в сфере обучения программированию,  $Kb^O$  – база знаний общего назначения DBpedia. Повышение квалификации агента обеспечивается тем, что на строго формализованному, созданному на основе моделей компетенций графу знаний  $G$ , сопоставляются постоянно обновляющиеся знания общего характера  $Kb^O$  и также обновляющиеся специализированные знания  $Kb^A$ .

Описанное обновление знаний требует существенно меньших временных затрат по сравнению с повышением квалификации человека. В условиях активного развития сферы программирования, когда версии языков и пакетов инструментов обновляются практически ежегодно, наличие

интеллектуальных агентов, знающих особенности конкретных инструментов, может быть существенным подспорьем для преподавателя и оставлять время для повышения квалификации в части фундаментальных знаний.

При этом использование графа знаний  $G$  позволяет снизить вероятность ошибок, допускаемых преподавателем при определении уровня компетенций ученика  $e_i^l$  при оперативной корректировке ИОТ. Во второй главе было показано, что снижение количества ошибок достигается путем использования в функции `search` модифицированного алгоритма Graph RAG [35].

Переходя к факторам и подэтапам методики, связанным с базой знаний  $Kb^B$ , необходимо заметить, что качество базы знаний и возможности ее использования агентом являются ключевой характеристикой предлагаемой методики.

База знаний  $Kb^B$  включает сегмент  $Kb^A$ , который формируется на основе дисциплин  $c_i$  путем применения функции `to_kb` к учебно-методическим материалам. Поэтому при определении параметров ИЭОС, включающем указание дисциплин, формировании основных и альтернативных сценариев ИОТ, формировании компетенций агентов и прохождении ИОТ важным является влияние фактора  $\varphi_8$  «Материал не понятен обучающимся». Попадание в сегмент  $Kb^A$  запутанного, нелогичного и плохо структурированного материала может снизить качество базы знаний. Однако алгоритм функции `to_kb` предусматривает проверку извлекаемых из  $e_i$  знаний по графу знаний  $G$  и базе знаний общего назначения  $Kb^O$ , что обеспечивает качество формирования сегмента  $Kb^A$ .

Использование базы знаний  $Kb^B$  снижает количество ошибок участников учебного процесса, что обусловлено следующим:

- 1) Интеллектуальный агент  $a_r^l$  при обработке входящих сообщений применяет функцию `search`, которая обращается для проверки информации к

базе знаний  $Kb$ . Это позволяет интеллектуальному агенту в режиме реального времени дать рекомендации обучающемуся.

2) Функция  $gen$ , которая генерирует ответы агента-преподавателя  $a_r^l$ , использует промпты, созданные функцией  $aug$ , обращающейся к базе знаний  $Kb$  для уточнения контекста.

В модели взаимодействия участников учебного процесса  $M^A$  было определено, что при прохождении каждой фактической ИОТ существует множество сообщений  $Txt_i$ . Текст каждого сообщения  $txt_i$  является входным поисковым запросом для функции поиска, которая описывается следующей функцией (99)

$$\text{search} (query, Kb^B, G), \quad (99)$$

где  $query$  – входной запрос на естественном языке,  $query = txt_i$ ;  
 $Kb^B$  – база знаний, описанная в модели взаимодействия участников учебного процесса;  
 $G$  – граф знаний, описанный в модели взаимодействия участников учебного процесса.

Основные шаги работы функции можно формализовать в виде схемы, приведенной на рисунке 15.

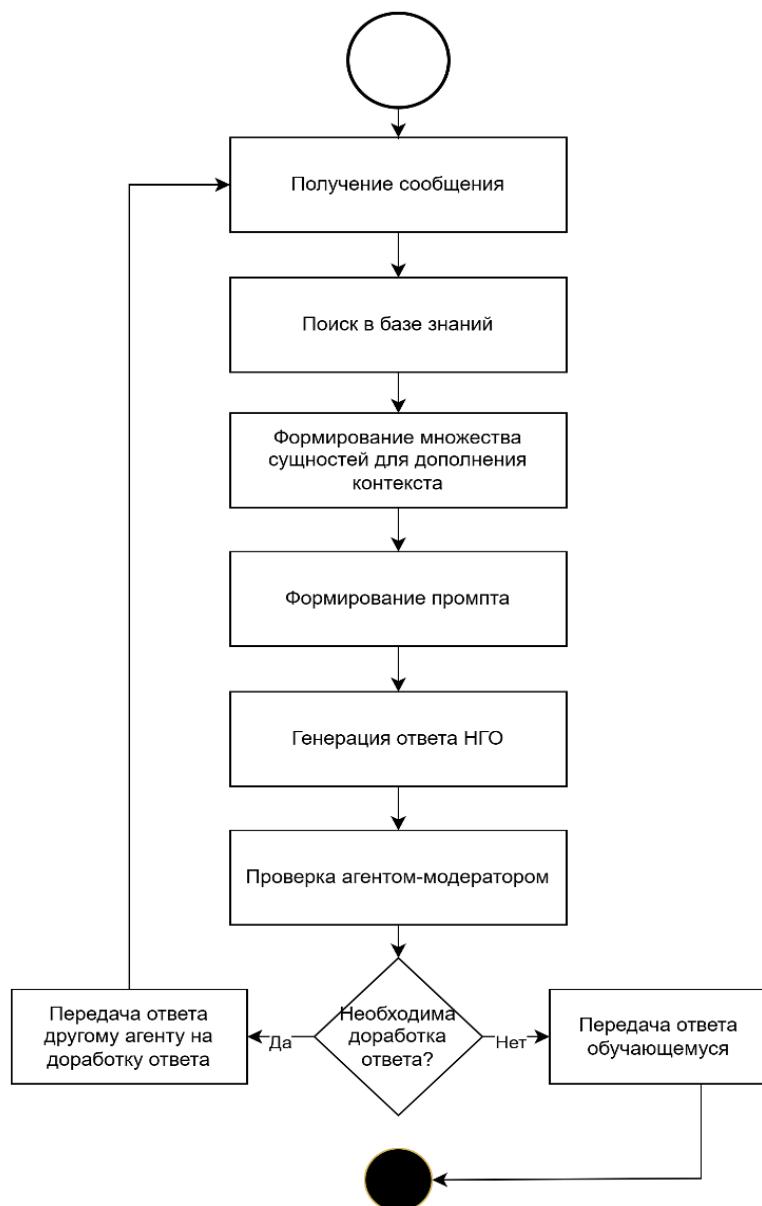
Функция  $\text{search}$  извлекает сущности  $subj$  из запроса  $query$ , производит поиск сущностей в базе знаний  $Kb^B$  и графе знаний  $G$ . Таким образом сущностям запроса  $query$  сопоставляются вершины графа знаний  $G$ , элементы базы знаний общего назначения  $Kb^O \subset Kb^B$  и специализированной базы знаний  $Kb^A \subset Kb^B$ . В результате образуется множество сущностей  $Subjs$ , которое могут быть описаны формулой (100)

$$Subjs = \{subj_1, subj_2, \dots, subj_n\}, \quad (100)$$

где  $subj_i$  является следующим кортежем, описываемым формулой (101)

$$subj_i = \langle fragment, r, kg \rangle, \quad (101)$$

где  $fragment$  – токен (часть текста) из запроса  $query$ ;  
 $r \in R^k$  – отношение из ранее описанного множества  $R^k$ ;  
 $kg$  – элемент базы знаний общего назначения  $Kb^O \subset Kb^B$ , специализированной базы знаний  $Kb^A \subset Kb^B$  или вершина графа знаний  $G$ .



Источник: составлено автором.  
Рисунок 15 – Основные шаги функции search

Таким образом справедливо следующее утверждение:  
 $kg \in Kb^0 \vee kg \in Kb^A \vee kg \in G$ . При этом соблюдается также условие  
 $r \in R^k, R^k = \{g_i \rightarrow kb_i^0\} \cup \{kb_i^A \rightarrow kb_i^0\}$ , поэтому извлеченные сущности  
 $subj_i$  являются только элементами базы знаний  $Kb^B$  или графа знаний  $G$ . Это  
позволяет проводить дополнение контекста только достоверными данными, а  
также выявить неявные отношения во входном запросе *query*.

Поиск проводится в два этапа: сначала на графе знаний  $G$ , после этого  
в базе знаний  $Kb^B$ .

Поиск в графе знаний  $G$  основан на алгоритме поиска в ширину и может  
быть описан следующим образом:

- 1) На основе роли агента  $a_i^L$  и сведений об обучающемся  $a_j^S$   
производится выбор подграфа  $G^1$ , в котором содержатся актуальные для  
участвующих агентов компетентностные(профессиональные) стандарты.
- 2) Для найденных подграфов  $G^1$  на основе роли агента  $a_i^L$  и сведений  
об обучающемся  $a_j^S$  определяются актуальные уровни квалификации.  
Актуальные уровни квалификации выделяются в подграф  $G^2$ .
- 3) Для каждого токена *fragment* в подграфе  $G^2$  проводится  
семантический поиск по вершинам, описывающим обобщенные трудовые  
функции. Найденные вершины выделяются в подграф  $G^3$ .
- 4) Для каждого токена *fragment* в подграфе  $G^3$  проводится поиск  
трудовых функций. Найденные трудовые функции выделяются в подграф  $G^4$ .
- 5) Между вершинами графа  $G^4$  и  $G^3$  устанавливаются связи,  
эквивалентные отношению «часть-целое». Шаг повторяется для пар  
подграфов  $G^3$  и  $G^2$ ,  $G^2$  и  $G^1$ . Полученный граф сохраняется.
- 6) Для данного токена *fragment* получен результат поиска  $kg \in G$ .

Поиск в базе знаний  $Kb^B$  может быть описан следующим образом:

- 1) Для каждого токена *fragment* проводится поиск в сегменте  $Kb^A$ ,  
найденные элементы сегмента сохраняются в множество  $Kba^1$ .

2) Для каждого элемента множества  $Kba^1$  выделяются элементы, непосредственно связанные с ним отношениями эти элементы добавляются в множество  $Kba^2$ .

3) Множества  $Kba^1$  и  $Kba^2$  объединяются в множество  $Kba^3$ .

4) Для каждого токена *fragment* проводится поиск в сегменте  $Kb^0$ , найденные элементы сегмента сохраняются в множество  $Kbo^1$ .

5) Для каждого элемента множества  $Kbo^1$  выделяются элементы, непосредственно связанные с ним отношениями эти элементы добавляются в множество  $Kbo^2$ .

6) Множества  $Kbo^1$  и  $Kbo^2$  объединяются в множество  $Kbo^3$ .

7) Производится поиск отношений  $r \in R^k$  между элементами множеств  $Kba^3$  и  $Kbo^3$ . Полученный результат, множества  $Kba^3$  и  $Kbo^3$  объединяются во временный сегмент базы знаний  $Kb^4$ .

8) Для данного токена *fragment* получен результат поиска  $kg \in Kb^0 \vee kg \in Kb^4$ .

В результате совместной работы алгоритмов поиска в графе знаний  $G$  и поиска в базе знаний  $Kb^B$  формируется множество  $Subjs$ , в котором токены сопоставлены подграфам и сегментам базы знаний.

Этап дополнения контекста заключается в создании текстового запроса, который отправляется большой языковой модели. В настоящей работе входной текст формируется функцией (102)

$$prompt_i = aug(txt_i, context^{pers}, context^{comp}, context^{crs}, templ), \quad (102)$$

где  $txt_i$  – текст сообщения, отправленного большой языковой модели;

$context^{pers}$  – контекст компетенций, который описывает персональные качества агента;

$context^{comp}$  – контекст компетенций, который описывает профессиональные компетенции агента;

$context^{crs}$  – контекст компетенций, который описывает компетенции агента по курсу;

$templ$  – шаблон, по которому происходит объединение  $txt_i, context^{pers}, context^{comp}, context^{crs}$  в единый текст. В шаблоне могут быть указана дополнительные фразы, изменен порядок следования входящих в него текстов или проведены иные преобразования.

Функция  $aug$  подставляет свои аргументы в шаблон  $templ$  и таким образом формирует входной текст, передаваемый НГО.

Этап генерации заключается в том, что входной текст, полученный функцией  $aug$  передается большой языковой модели. Как было показано при формализации модели взаимодействия участников учебного процесса, агент на основе большой языковой модели описывается как  $\langle \pi, mdl, ctx \rangle$ .

Полученный на предыдущем шаге результат вычисления функции  $aug$ , в которую были переданы  $context^{pers}, context^{comp}$  агента  $a_i^L$  передаются модели  $mdl$  для генерации сообщения, которое будет передано обучающемуся. Генерацию сообщения можно описать формулой (103)

$$txt_{i+1} = gen(prompt_i, mdl), \quad (103)$$

где  $txt_{i+1}$  – итоговый текст, сгенерированный моделью  $mdl$  в терминах  $M^A$ , является сообщением  $txt_{i+1}$  в множестве сообщений  $Txt$  для прохождения  $d_i$  ИОТ.

В связи с этим сообщение  $txt_{i+1}$  передается агенту-модератору  $a_0^M \in A^M$ , который может передать его другому агенту  $a_j^L$  для дополнительной обработки или агенту-верификатору для проверки (например, проверки правильности сгенерированного программного кода).

Как было сказано выше, на вход функции  $aug$  подаются контексты  $context^{pers}$ ,  $context^{comp}$ ,  $context^{crs}$ , каждый из которых формируется из собственного источника.

Контекст  $context^{pers}$  описывающий роль интеллектуального агента, задается в настройках ИЭОС  $B$ . Обзор литературы показал, что задание роли повышает эффективность работы НГО, а в случае многоагентной системы является необходимым, поскольку роль определяет функции агента и его отличия от прочих агентов.

Контекст  $context^{comp}$  формируется из информации, содержащейся в графе знаний  $G$ , который состоит из подграфов  $g_i$ , соответствующих моделям компетенций.

Контекст  $context^{crs}$  строится на основе базы знаний  $Kb^B$ , при этом часть, относящаяся к конкретному учебному модулю  $c_i$  содержится в сегменте базы знаний  $Kb^A$ , общепрофессиональные знания содержатся в сегменте  $Kb^O$ , а наличие между этими сегментами связей  $R^k$  обеспечивает актуальность учебных модулей и связность знаний в базе знаний.

Указанные контексты являются текстовыми, поэтому необходимо описать функции, которые могут преобразовать данные графа знаний  $G$  и базы знаний  $Kb^B$  в текстовый вид. При этом преобразование проводится над элементами полученного с помощью функции  $search$  множества  $Subjs$ , элементы которого  $subj_i$  включают  $kg_i$ , такие, что  $kg \in Kb^O \vee kg \in Kb^A \vee kg \in G$ . Иными словами, преобразование в текст проводится над подграфами, включающими понятия базы знаний, и вершина графа знаний, связанные отношения с терминами, выделенными из текста запроса  $query$ .

Для преобразования необходимо формализовать взаимодействие интеллектуального агента  $a_i^L$  с базой знаний  $Kb^B$  и  $G$  как кортеж (104)

$$\langle a_i^L, Subjs, G, Kb^B, ctx^{comp}, ctx^{crs}, g\_to\_txt, kb\_to\_txt \rangle, \quad (104)$$

где  $a_i^L$  – интеллектуальный агент;  
 $Subjs$  – результат работы функции search;  
 $G$  – граф знаний;  
 $Kb^B$  – база знаний;  
 $context^{comp}$  – предложенный в модели  $M^B$  контекст компетенций, который описывает профессиональные компетенции агента;  
 $context^{crs}$  – предложенный в модели  $M^B$  контекст компетенций, который описывает профессиональные компетенции агента;  
 $g\_to\_txt$  – функция преобразования для  $subj_i, kg_i$  которых удовлетворяют условию  $kg_i \in G$  можно описать формулой (105)

$$context^{comp} = g\_to\_txt(Subjs), \quad (105)$$

где  $Subjs$  – подмножество  $Subjs$ , для которого выполняется условие  $kg_i \in G$ .

Алгоритм работы функции основывается на обходе графа в глубину и может быть описан следующим образом:

- 1) В множестве  $Subjs$  выбираются элементы, которые содержат вершины из подграфа  $G^1$ , описанного в алгоритме поиска.
- 2) Для каждой выбранной вершины  $g_j^1$  производится обход в глубину, при котором текст, содержащийся в пройденных вершинах и ребрах, соединяющих их, добавляется к результату работы функции.
- 3) Возвращается текстовое представление  $Subjs$ , полученное путем обхода.

$kb\_to\_txt$  – функция преобразования для  $subj_i, kg_i$  которых удовлетворяют условию  $kg \in Kb^O \vee kg \in Kb^A$  можно описать формулой (106)

$$context^{crs} = kb\_to\_txt(Subjskb), \quad (106)$$

где  $Subjskb$  – подмножество  $Subjs$ , для которого выполняется следующее условие  $kg \in Kb^0 \vee kg \in Kb^A$ .

Алгоритм работы функции основывается на обходе графа в глубину и может быть описан следующим образом:

- 1) в множестве  $Subjskb$  выбираются элементы  $kg_j^{kba}$ , для которых справедливо условие  $kg \in Kb^A$ ;
- 2) свойства и отношения выбранных элементов  $kg_j^{kba}$  преобразуются в текст, который добавляется к результатам;
- 3) для  $kg_j^{kba}$  в текст преобразуются найденные на этапе поиска отношения  $r \in R^k$ , через которые установлена связь с элементами  $kg_j^{kbo}$  множества  $Subjskb$ , для которых справедливо условие  $kg \in Kb^0$ ;
- 4) свойства и отношения оставшихся элементов  $kg_j^{kbo}$  преобразуются в текст;
- 5) возвращается текстовое представление  $Subjskb$ .

Таким образом применение модифицированного алгоритма генерации с дополненной выборкой позволяет интеллектуальным агентам взаимодействовать с обучающимися на каждом этапе  $tr_i$  прохождения ИОТ. Модифицированный алгоритм генерации с дополненной выборкой отличается в части реализации следующих дополнений:

- 1) Реализован оригинальный алгоритм поиска  $search(query, Kb^B, G)$ , который проводит извлечение сущностей в графе знаний и связанной с ним базе знаний.
- 2) Реализован оригинальный алгоритм дополнения  $aug(txt_i, ctx^{pers}, ctx^{comp}, ctx^{crs}, templ)$  и вспомогательные функции  $g\_to\_txt(Subjs)$  и  $kb\_to\_txt(Subjskb)$ , преобразующие найденные с помощью функции  $search$  подграфы и сегменты баз знаний в текстовые

сообщения с учетом модели компетенций, заданной профессиональными стандартами.

Использование достоверной информации из графа знаний  $G$  и базы знаний  $Kb^B$  повышает качество генерации, снижая вероятность ошибок в сгенерированном тексте, что позволяет снизить влияние деструктивных факторов  $\Phi_s$ .

Необходимо определить критерии достижения заданного уровня освоения модулей учебных дисциплин при электронном обучении программированию с соблюдением предусмотренных образовательным стандартом требований с учетом влияния результатов алгоритма генерации с дополненной выборкой.

Сегменты базы знаний  $Kb^B$  формируются различным образом. Сегмент  $Kb^O$  – это готовая база знаний в формате OWL, его формирование заключается в копировании актуальной версии из DBpedia. Сегмент  $Kb^A$  формируется для каждой описанной в модели  $M^A$  дисциплины  $c_i$  путем применения функции  $to\_kb$  к учебно-методическим материалам  $e_i \in E_{c_i}$ .

Работу функции  $to\_kb$  можно описать в виде следующего алгоритма:

1) Все  $data$  данного  $e_i$  разделяются на термы, для которых проводится поиск семантически близких вершин в графе знаний  $G$ . Термы, для которых найдены соответствующие вершины запоминаются в множество  $TG$ .

2) Все  $data$  данного  $e_i$  разделяются на термы, для которых проводится поиск соответствующих понятий в сегменте базе знаний общего назначения  $Kb^O$ . Термы, для которых найдены соответствующие вершины запоминаются в множество  $TK$ .

3) Из множеств  $TG$  и  $TK$ , а также отношений, связывающих их с графом знаний  $G$  и сегментом базы знаний общего назначения  $Kb^O$  формируется сегмент базы знаний  $Kb^A$ .

Таким образом множество отношений  $R^k$ , установленных между элементами  $Kb^A, G$  и  $Kb^O$  может быть описано формулой (107)

$$R^k = \{g_i \rightarrow kb_j^O\} \cup \{kb_i^A \rightarrow kb_j^O\}, \quad (107)$$

где  $g_i$  – вершина графа знаний  $G$ ;

$kb_j^O$  – элемент базы знаний общего назначения  $Kb^O$ ;

$kb_i^A$  – элемент специализированная база знаний  $Kb^A$ ;

Стрелкой  $\rightarrow$  обозначено отношение между элементами баз знаний и графиком знаний.

Множество  $R^k$  формируется следующим образом: для каждого элемента  $kb_i^A$  устанавливается отношение  $R$  хотя бы с одной вершиной графа знаний  $G$  в связи с тем, что,  $g_i = convert^g(s_i), s_i \leftrightarrow e_i$ . Выявление отношений  $R$  между  $g_i$  и  $kb_i^O$  производится с помощью предложенного F. Osborne и E. Motta алгоритма Klink-2 [117].

Модифицированный алгоритм работает со структуризованными источниками данных, например DBpedia и слабоструктуризованными источниками, например GitHub. Для его реализации могут применяться средства обработки естественного языка, применяемые в работе со структуризованными источниками данных [56], средства трансляции запросов в промежуточные представления [55] или специализированные алгоритмы извлечения данных из баз знаний [54]. Необходимо заметить, что использование онтологического подхода позволяет в автоматическом режиме управлять вычислениями в продукционной системе приобретения знаний из текстов: активировать систему продукции, строить соответствующую активной системе продукции рабочую память, заполнять ее фактами и запускать машину логического вывода [53].

Частная модель построения контекста интеллектуального агента на основе базы знаний адаптирована для использования в действиях

«Формирование ИОТ», «Оценка текущих параметров ИОТ» и «Корректировка ИОТ» методики для вычисления функций  $rtr(tr_i, A)$ ,  $rs(tr_i, a^s, \Phi_s)$  и  $rl(tr_i, a^l, \Phi_l)$ .

### **Выводы по третьей главе**

В третьей главе предложена оригинальная методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию для достижения заданного уровня знаний, умений и навыков за регламентированное время с применением основанных на нейросетях глубокого обучения интеллектуальных агентов и марковского процесса принятия решений, **отличающаяся от аналогичных методик Э.В. Мищенко [67], В. Д. Лиференко [64], Т. Wang [131], Р. Denny [98; 99], М.А. Косоноговой [12] тем, что:**

- на основании зависимости результатов участников учебного процесса на индивидуальном и групповом уровнях от влияния деструктивных факторов и учета текущего уровня знаний, умений и навыков определяется оптимальный по критерию достижения заданного уровня ЗУН за оставшееся время следующий этап ИУП, который должен быть включен в ИОТ;
- при этом индивидуальная образовательная траектория оперативно формируется и динамически корректируется с применением марковского процесса принятия решений, где функция полезности вычисляется НГО-агентами на индивидуальном уровне с последующим преобразованием на групповом уровне модели взаимодействия участников учебного процесса;
- НГО-агенты используют адаптированную частную модель построения контекста на основе двухсегментной базы знаний, компетентностный сегмент которой формируется с помощью генерации с дополненной выборкой на основе учебно-методических материалов.

Это позволяет путем применения методики получить оптимальный уровень ЗУН за регламентированное время путем перераспределения времени

участников учебного процесса и оперативного реагирования на влияние деструктивных факторов.

## Глава 4

### **Исследование эффективности методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию**

#### **4.1 Постановка задачи вычислительного эксперимента с использованием методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию**

Для подтверждения эффективности предложенной в третьей главе методики оперативного формирования ИОТ при электронном обучении программированию был проведен вычислительный эксперимент, состоящий из двух этапов:

- проверка эффективности методики оперативного формирования ИОТ при электронном обучении программированию;
- проверка эффективности применения частной модели построения контекста интеллектуального агента на основе двухсегментной базы знаний с применением алгоритма генерации с дополненной выборкой.

Для проведения эксперимента был выбран курс «Структуры и алгоритмы обработки данных», который изучается студентами направления «Прикладная информатика» Восточно-Сибирского государственного университета технологий и управления. В рамках эксперимента сформированы две независимые группы по 30 студентов, отобранных случайным образом из генеральной совокупности студентов ( $n = 1200$ ).

Эксперимент проводился с применением разработанного автором прототипа, комплекса программ, реализующих описанную в параграфе 4.2 архитектуру ИЭОС *optimus(B)*.

На каждом этапе первая группа студентов работала только с преподавателем, вторая группа студентов работала с преподавателем и интеллектуальными агентами, предусмотренными методикой оперативного формирования ИОТ при электронном обучении программированию.

Цель эксперимента заключалась в проверке эффективности методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию. Эффективность может быть оценена по двум показателям:

- уровень ЗУН, который измеряется введенной в параграфе 2.2 функцией группового уровня  $rtr(tr_i, A)$ ;

- фактическое время прохождения ИОТ, измеряемое введённой в частной модели распределения времени введенной в параграфе 2.3 формулой  $t_{\text{Факт}} = t_{\text{Факт}}^{\text{Конт}} + t_{\text{Факт}}^{\text{СРС}} + t_{\text{Факт}}^{\Delta} + t_{\text{Факт}}^{\text{СРП}}$ .

Для подтверждения эффективности использования методики оперативного формирования ИОТ сформулированы следующие допущения:

- нулевая гипотеза  $H_0$  – использование методики оперативного формирования ИОТ в обучении программированию не оказывает значимого влияния на достижение требуемого уровня освоения профессиональных компетенций в заданные сроки обучения;
- альтернативная гипотеза  $H_1$  – использование методики оперативного формирования ИОТ способствует достижению требуемого уровня освоения профессиональных компетенций в заданные сроки обучения.

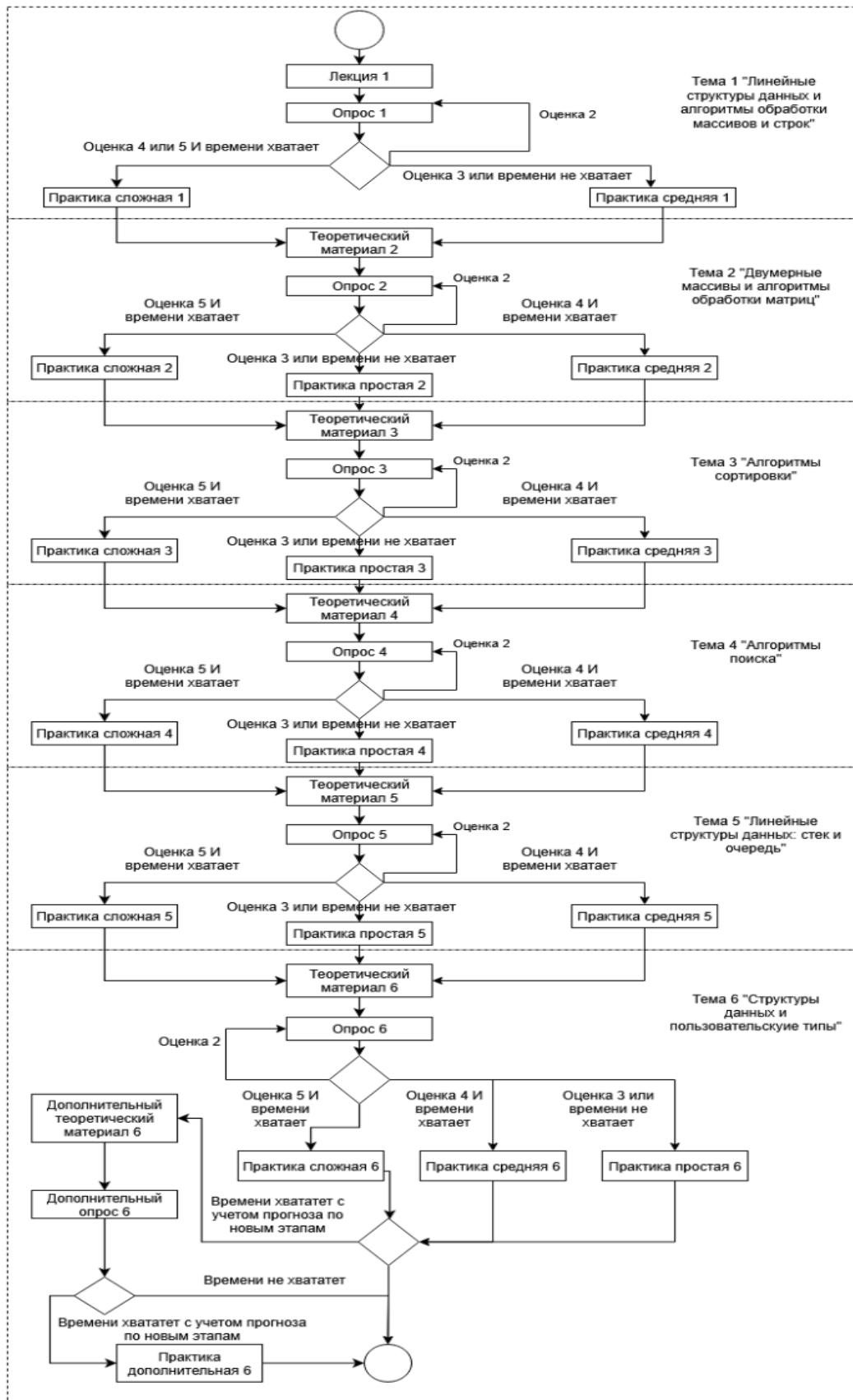
Подтверждение гипотезы  $H_1$  должно продемонстрировать, что комплекс программ, разработанный по методике оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию с применением НГО способен обеспечить достижение заданного уровня ЗУН путем взаимодействия обучающегося, преподавателя и интеллектуальных агентов с заданными ролями и базой знаний  $Kb$ ,

полученной путем преобразования учебно-методических материалов путем применения алгоритма генерации с дополненной выборкой.

**Описание вычислительного эксперимента по проверке эффективности методики оперативного формирования ИОТ при электронном обучении программированию.** Исходными данными первого этапа эксперимента являются следующие темы  $q_i$  сокращенного курса «Структуры и алгоритмы обработки данных»:

- 1) «Линейные структуры данных и алгоритмы обработки массивов и строк».
- 2) «Двумерные массивы и алгоритмы обработки матриц».
- 3) «Алгоритмы сортировки».
- 4) «Алгоритмы поиска».
- 5) «Линейные структуры данных: стек и очередь».
- 6) «Структуры данных и пользовательские типы».

Индивидуальный учебный план курса «Структуры и алгоритмы обработки данных» представлен на рисунке 16. План курса включает шесть тем: первая тема содержит лекцию, опрос и практические занятия двух уровней сложности, в темах 2–5 вариативность расширяется за счет применения практики трех уровней сложности, а в последней 6 теме предлагается дополнительный теоретический материал. Это позволяет реализовать описанную в методике стратегию достижения обучающимся минимально удовлетворительного уровня ЗУН  $R^{\min}$  при приближении разности планового и фактического времени к порогу риска или максимизации уровня ЗУН  $R^{\max}$  в противном случае.



Источник: составлено автором.

Рисунок 16 – Визуальное представление индивидуального учебного плана курса «Структуры и алгоритмы обработки данных»

Плановые значения времени каждого этапа заданы в соответствии с таблицей 7.

Таблица 7 – Плановое распределение времени по этапам

Тема	Этап	СРС в часах	Контактная работа в часах	СРП в часах	Всего в часах
Линейные структуры данных и алгоритмы обработки массивов и строк	Лекция 1	2,5	4,5	6,25	13,25
	Опрос 1	1,5625	0,5	1,5	3,5625
	Практика 1	5	5	5	15
Двумерные массивы и алгоритмы обработки матриц	Лекция 2	2,5	4,5	6,25	13,25
	Опрос 2	1,5625	0,5	1,5	3,5625
	Практика 2	5	5	5	15
Алгоритмы сортировки	Лекция 3	3	5,5	7,8	16,3
	Опрос 3	1,875	0,5	2	4,375
	Практика 3	6	6	6	18
Алгоритмы поиска	Лекция 4	2,5	4,5	6,25	13,25
	Опрос 4	1,5625	0,5	1,5	3,5625
	Практика 4	5	5	5	15
Линейные структуры данных: стек, очередь	Лекция 5	2,5	4,5	6,25	13,25
	Опрос 5	1,5625	0,5	1,5	3,5625
	Практика 5	5	5	5	15
Структуры данных и пользовательские типы	Лекция 6	3	5,5	7,8	16,3
	Опрос 6	1,875	0,5	2	4,375
	Практика 6	6	6	6	18
	Доп. Теория 6	3	5,5	7,8	16,3
	Доп. Опрос 6	1,875	0,5	2	4,375
	Доп практика 6	6	6	6	18

Источник: составлено автором.

На каждом этапе в первой группе формировались описанные в параграфе 2.2 индивидуальные функции студента *rs* и преподавателя *rl*; во второй группе дополнительно учитывался вклад агента *rl*, при нулевой самостоятельной работе и пониженной чувствительности к деструктивным факторам.

Процедура переходов между видами практик и добавления дополнительного модуля была формализована с опорой на схему алгоритма прохождения ИОТ, которая показана на рисунке 16. Схема отражает

последовательность: «Лекция» → «Опрос» → «Практика» в каждой из шести тем. Выбор варианта практики в теме осуществлялся по правилу достаточности: при достижении требуемой оценки опроса и наличии временного ресурса назначались задания повышенной сложности; при недостатке времени или недостижении порога выбирался более простой вариант. Такой механизм обеспечивал совместный учёт качества и ресурсных ограничений. Для шестой темы дополнительно предусмотрена развилка на вспомогательный модуль с отдельными узлами контроля времени, что обеспечивает согласование выбора траектории с ограничениями  $T^{max}$ . Такой подход обеспечивает оптимальный по критерию достижения заданного уровня ЗУН за оставшееся время выбор следующего этапа ИОТ.

На каждом этапе фиксировались фактические значения времени самостоятельной работы  $t^{CPC}$ , контактного времени с преподавателем  $t^{Конт}$  и времени самостоятельной работы преподавателя  $t^{CPN}$ . Дополнительно рассчитывались векторы  $rs$  и  $rl$ , интегральная оценка  $ctrl$  и накопленный результат  $rtr$ . Полученные результаты эксперимента первой группы в приложении Д, второй группы в приложении Е.

В таблице 8 представлены агрегированные данные о времени контактной и самостоятельной работы, времени работы преподавателя и значению уровня ЗУН по группе, работавшей без применения методики оперативного формирования ИОТ.

Таблица 8 – Сводка по студентам: первая группа

Студент	$\Sigma t^{CPC}$ в часах	$\Sigma t^{CPN}$ в часах	$rtr$ в баллах	$\Sigma t^{Конт}$ в часах
1	2	3	4	5
1	54	85,5	66,325	65
2	65,5	90	63,8	70
3	52,5	87,5	65,725	68,5
4	50	84,5	67,125	63,5
5	60	78,5	65,925	67,5
6	55	76	67	69
7	67,5	88	79,45	68,5
8	61	76	65,325	71

## Продолжение таблицы 8

1	2	3	4	5
9	55	70,5	67,25	65
10	65	84	81,275	58
11	62,6	93,1	73,611	65,0
12	58,8	82,3	70,134	66,1
13	63,5	84,2	68,375	62,6
14	68,9	74,6	67,238	62,2
15	58,2	80,3	60,019	69,9
16	58,2	84,5	64,67	72,0
17	69,2	76,4	66,26	66,5
18	64,3	86,2	75,565	70,7
19	56,7	79,9	71,19	68,2
20	62,9	81,9	58,275	64,3
21	56,8	79,9	71,071	68,2
22	56,8	95,6	66,723	72,7
23	61,1	83,7	64,934	66,7
24	47,9	77,0	72,834	72,8
25	49,1	89,0	75,404	56,7
26	56,2	75,9	74,793	70,0
27	53,4	85,1	63,939	67,1
28	61,5	71,2	67,188	65,7
29	54,1	75,2	71,115	67,2
30	51,0	85,0	75,064	59,2
Среднее	58,55	82,05	68,92	66,66

Источник: составлено автором.

В таблице 9 представлены агрегированные данные о времени контактной и самостоятельной работы, времени работы преподавателя и значению уровня ЗУН по группе, работавшей с применением методики оперативного формирования ИОТ.

Таблица 9 – Сводка по студентам: вторая группа

Студент	$\Sigma t^{\text{CPC}}$ в часах	$\Sigma t^{\text{CРП}}$ в часах	rtr в баллах	$\Sigma t^{\text{КонтПреп}}$ в часах	$\Sigma t^{\text{КонтАгент}}$ в часах
1	2	3	4	5	6
1	52,5	86	70,433	41	17,5
2	54	84	69,617	46,5	19
3	67,5	80,5	70,133	36,5	15
4	62,5	74	70,3	50,5	20
5	57,5	83	71,067	39,5	14,5

Продолжение таблицы 9

1	2	3	4	5	6
6	60	82	69,017	45	19
7	58	74	70,917	44,5	17,5
8	61,5	74,5	69,867	42,5	17,5
9	58	77	70,517	47,5	18,5
10	54	75,5	71,217	45	17,5
11	57,7	72,4	70,929	44,6	15,3
12	60,3	76,9	69,767	49,0	18,4
13	65,4	77,3	71,347	37,1	19,0
14	56,3	75,2	69,43	44,4	15,6
15	55,0	78,1	70,789	44,7	18,7
16	56,4	80,7	71,885	46,9	17,7
17	62,8	87,5	69,711	38,6	18,4
18	60,2	79,6	70,001	38,3	20,3
19	56,2	80,0	70,456	45,8	16,6
20	61,0	78,5	70,044	44,9	15,7
21	59,1	70,1	69,329	44,7	15,5
22	63,1	78,7	70,435	45,1	15,6
23	55,5	79,1	69,662	40,9	16,9
24	57,2	90,1	70,712	44,6	17,6
25	56,9	78,0	69,76	44,9	17,5
26	52,0	80,2	71,447	40,7	18,4
27	60,0	78,7	69,853	51,3	17,0
28	59,8	73,5	70,168	45,6	19,5
29	58,7	84,1	70,944	38,8	16,6
30	57,6	82,3	69,547	46,3	21,7
Среднее	58,55	79,05	70,31	43,85	17,6

Источник: составлено автором.

В рамках эксперимента в качестве генеральной совокупности рассматривались студенты направления «Прикладная информатика» ( $n = 1200$ ), проходившие обучение по курсу «Структуры и алгоритмы обработки данных». Из генеральной совокупности методом случайного отбора сформированы две репрезентативные группы по 30 человек.

В таблице 10 приведено время контактной работы  $t_{\text{Факт}}^{\text{Конт}}$  для группы, работавшей без применения методики оперативного формирования ИОТ. Среднее значение времени контактной работы составляет от 9,6 до 10,7 часов на тему.

Таблица 10 – Распределение времени контактной работы по темам: первая группа

В часах

Тема	Тема 1	Тема 2	Тема 3	Тема 4	Тема 5	Тема 6
1	11	9,5	12,5	8	10,5	13,5
2	9	9	15	13	12,5	11,5
3	11	12	14	12	8	11,5
4	10,5	9,5	10,5	7,5	11,5	14
5	10	13	13,5	7,5	10	13,5
6	15	11	12,5	9,5	8,5	12,5
7	9,5	8	9,5	12,5	8,5	20,5
8	10	12,5	14	9,5	14,5	10,5
9	9,5	14	11	10	7,5	13
10	8,5	8	5	12,5	4,5	19,5
11	11,3	7,6	7,9	9,5	9,8	9,1
12	10,2	13,7	12,4	10,9	8,8	15
13	11,6	10,1	14	12,3	9,9	14,9
14	13,2	10,7	12,3	9,2	4	14
15	10	7,5	11,5	9,8	9	13,2
16	10	9,4	10,9	7,8	10,6	9,3
17	13,3	10,8	7,5	7,6	13,8	12,6
18	11,8	8,1	9,7	12	8,1	12,9
19	9,6	11,4	10,4	13,1	7,3	11,3
20	11,4	9,3	14,9	10	8,2	13,5
21	9,6	10	12,8	12,4	12,2	15,4
22	9,6	9,3	6,6	11	10,5	20,3
23	10,8	14,6	12,8	8,8	8,1	14,6
24	6,9	10,6	10,7	11	11,1	14,9
25	7,3	8,3	9,8	13,5	9,9	13,8
26	9,4	12,4	13,6	10,1	12,3	7,6
27	8,6	8	14,8	13,6	7,6	13,9
28	11	11	14,5	4,5	8,7	14,2
29	8,8	6,4	9,3	12	8,5	22,3
30	13,6	23,8	18,6	4,9	13,6	17,2
Среднее по теме	10,4	10,65	11,75	10,2	9,6	14

Источник: составлено автором.

В таблице 11 приведено время контактной работы для группы, работавшей с применением методики оперативного формирования ИОТ. Среднее значение времени контактной работы снижается до 6,3–8,3 часов.

Таблица 11 – Распределение времени контактной работы по темам: вторая группа

В часах

Тема	Тема 1	Тема 2	Тема 3	Тема 4	Тема 5	Тема 6
1	8	5	6,5	6,5	6,5	8,5
2	8	6	7,5	5	8	12
3	4,5	6,5	4,5	7,5	7,5	6
4	7,5	10,5	8,5	7,5	8	8,5
5	6,5	6	6	4	8	9
6	9,5	7,5	8	5	7,5	7,5
7	8	8	8,5	7,5	5,5	7
8	6,5	8	5,5	6	8,5	8
9	5,5	7,5	10,5	7,5	7,5	9
10	6	6	11,5	6,5	7	8
11	6,7	7,8	6,2	5,3	7	13,6
12	7,4	5,7	8,2	6,2	8,1	9,3
13	6,9	9,5	8,3	6,7	7,6	10,2
14	5,3	5,9	6,1	6,7	6,3	9,9
15	8,7	6,6	11,8	7,4	7,6	9,4
16	8,1	8,4	8,7	6,3	7,7	7,9
17	8,2	5,2	5,1	8,2	6,6	9,6
18	5,7	7,5	9,1	6	7,5	7,2
19	9,1	9,1	5,6	9,8	7,5	8
20	4,9	4,6	9,4	7,1	6,4	7,6
21	7,9	7,4	10,2	5,2	7,7	8,5
22	10,2	7,5	5,9	4,9	7,9	12,1
23	5,5	8,3	9,8	6,9	8,3	5,4
24	6,2	5,2	8,6	6	8,3	9,5
25	7,1	5	9,5	7,2	6,2	5,8
26	6,3	7,9	11,9	6,9	6,6	7,7
27	4,7	7,6	7,2	6,2	7,9	10,1
28	7,1	7,5	6,1	5,2	7,8	8,5
29	5,4	7,6	5,8	4,4	7,9	6,7
30	8,6	7,7	0,5	3,4	7,1	0
Среднее по теме	7	7,1	7,7	6,3	7,4	8,35

Источник: составлено автором.

При анализе средних значений контактного времени по темам также выявлены устойчивые различия. В первой группе наибольшая нагрузка приходилась на темы 3 и 6, где средние значения достигали 11,75 и 14 часов

соответственно. Во второй группе максимальные значения наблюдались в пределах 8,35 часов (тема 6), что существенно ниже показателей первой группы.

Для определения влияния применения методики оперативного формирования ИОТ на временные затраты был выполнен сравнительный анализ продолжительности контактной работы студентов с преподавателем и с агентом. Оценка снижения времени осуществляется по формулам (108) и (109)

$$\Delta t^{\text{Конт}} = \overline{t_1^{\text{Конт}}} - \overline{t_2^{\text{Конт}}}, \quad (108)$$

$$\omega^{\text{Конт}} = \frac{\Delta t^{\text{Конт}}}{\overline{t_1^{\text{Конт}}}} \times 100\%, \quad (109)$$

где  $\overline{t_1^{\text{Конт}}}$  – среднее время контактной работы в первой группе;

$\overline{t_2^{\text{Конт}}}$  – среднее время контактной работы преподавателя

во второй группе;

$\Delta t^{\text{Конт}}$  – абсолютное сокращение контактного времени;

$\omega^{\text{Конт}}$  – относительное сокращение контактного времени в процентах.

Расчёт доли взаимодействия с агентом выполнялся только второй группы по формуле (110)

$$r = \frac{\overline{t_2^{\text{Агент}}}}{\overline{t_2^{\text{Конт}}} + \overline{t_2^{\text{Агент}}}} \times 100\%, \quad (110)$$

где  $\overline{t_2^{\text{Агент}}}$  – среднее время контактной работы с агентом;

$\overline{t_2^{\text{Конт}}}$  – среднее время контактной работы с преподавателем;

$r$  – доля участия агента в контактной работе.

Для количественной оценки влияния методики оперативного формирования ИОТ на общее время контактной работы был рассчитан

показатель относительного сокращения общего контактного времени. Расчёт показателя осуществляется по формуле (111)

$$\omega^{\Sigma\text{Конт}} = \frac{\overline{t1\text{Конт}} - (\overline{t2\text{Агент}} + \overline{t2\text{Преп}})}{\overline{t1\text{Конт}}} \times 100\%. \quad (111)$$

На основании данных из таблиц 8 и 9 и вычислений получены следующие результаты:  $\overline{t1\text{Конт}} = 66,66$  ч.,  $\overline{t2\text{Конт}} = 43,85$  ч.,  $\Delta t\text{Конт} = 22,81$  ч.,  $\omega\text{Конт} = \frac{22,81}{66,66} \times 100 = 34,2\%$ ,  $\overline{t2\text{Агент}} = 17,6$  ч.,  $\overline{t2\text{Преп}} = 43,85$  ч.,  $r = \frac{17,6}{43,85+17,6} \times 100 = 28,64\%$ ,  $\omega^{\Sigma\text{Конт}} = 7,8\%$ .

Таким образом можно сделать вывод, что при сохранении среднего времени самостоятельной работы студентов (58,55 часов) и уровне сформированности ЗУН, сопоставимом с первой группой, время контактной работы с преподавателем сократилось на 34,2%. При этом суммарное контактное время преподавателя и агента уменьшилось на 7,8%, а доля взаимодействия с агентом составила 28,6%. Таким образом введенное в параграфе 2.3 фактическое время прохождения ИОТ  $t\text{Факт}$  уменьшается за счет уменьшения  $t\text{Конт}$ . Это подтверждает эффективность применения методики в модулях повышенной сложности в части достижения обучающимся требуемого уровня знаний, умений и навыков в установленные образовательным стандартом сроки при обучении программированию в интеллектуальной электронной образовательной системе путем оперативной корректировки индивидуальной образовательной траектории с учетом недетерминированности процесса обучения.

На основании этого можно сделать выводы, что при применении методики оперативного формирования ИОТ:

1) не снижается достигаемый уровень ЗУН, что подтверждается тем, что накопленная оценка  $rtr$  и распределение  $ctrlr$  остаются на уровне первой группы;

2) снижается время контактной работы обучающегося и преподавателя с одновременным перераспределением в пользу самостоятельной работы обучающегося во взаимодействии с интеллектуальными агентами, что обеспечивает более эффективное распределение времени преподавателя;

3) снижается зависимость достижения заданного уровня ЗУН от влияния деструктивных факторов из групп частоты возникновения ошибок  $\Phi_s^{as}$  и  $\Phi_l^{al}$  в связи с перераспределением времени самостоятельной работы студента  $t_{\text{Факт}}^{\text{CPC}}$  в пользу времени работы с интеллектуальными агентами.

Сравнительный анализ показал, что применением НГО-агентов и численного метода построения оптимальной ИОТ, реализованных в комплексе программ позволило снизить влияние деструктивных факторов на прохождение ИОТ и уменьшить фактическое время прохождения в среднем на 10–28%, достигая в отдельных темах 38%. Таким образом, применение методики оперативного формирования ИОТ при обучении программированию позволяет достигнуть заданный уровень ЗУН за регламентированное время с учетом зависимости выбора стратегии от приближения разности фактического и регламентированного времени освоения на текущем этапе ИОТ к порогу риска.

**Описание вычислительного эксперимента по проверке влияния контекста интеллектуальных агентов на эффективность методики оперативного формирования ИОТ при электронном обучении программированию.** Анализ результатов эксперимента показал, что эффект от применения методики оперативного формирования ИОТ при обучении программированию в значительной мере определяется воздействием деструктивных факторов  $\varphi_8, \varphi_{10}, \varphi_{13}, \varphi_{14}, \varphi_{15}$ . Воздействие данных факторов согласно модели взаимодействия участников учебного процесса корректируется путем применения НГО-агентов, контекст которых формируется на основе базы знаний, созданной из учебно-методических материалов курсов.

Построение контекста используемых в методике интеллектуальных агентов осуществляется с применением описанной в параграфе 2.3 частной модели построения контекста интеллектуального агента.

Для проведения эксперимента по проверке влияния контекста интеллектуальных агентов на эффективность методики оперативного формирования ИОТ при электронном обучении программированию двум группам студентов было предложено смоделировать взаимодействие по решению практических задач по 6 различным сценариям для решения шести практических задач из курса «Структуры и алгоритмы обработки данных» в рамках каждого сценария.

Как было показано в параграфе 2.2 интеллектуальный агент представляется в виде совокупности стратегии  $\pi$ , нейронной сети глубокого обучения  $mdl$  и контекста  $ctx$ . При таком подходе наиболее значимым существенно варьируемым компонентом интеллектуального агента становится контекст в связи с тем, что стратегия  $\pi$  определена в модели взаимодействия участников учебного процесса и может корректироваться только за счет коэффициента дисконтирования и функции корректировки ожидаемой полезности.

В настоящей диссертации не рассматриваются вопросы создания и оптимизации нейросетей глубокого обучения, а современные подходы к использованию моделей машинного обучения предполагают унификацию доступа к ним. Поэтому для проведения эксперимента целесообразно выбрать какую-либо широко распространенную НГО.

В качестве таких НГО в эксперименте для работы интеллектуальных агентов были использованы широко известные модели – так называемые ChatGPT и GigaChat. В ходе эксперимента учитывались такие метрики как среднее количество исправлений обучающимся кода, среднее количество найденных проблем в коде, среднее количество рекомендаций от агентов,

количество итераций до верного решения и процент понятных обучающемуся исправлений.

Первая группа студентов работала только с преподавателем, вторая группа студентов работала с преподавателем и интеллектуальными агентами, предусмотренными моделью взаимодействия участников учебного процесса. При работе с интеллектуальными агентами студенты должны были следовать сценариям взаимодействия, приведенным в приложении В.

Каждый студент предварительно самостоятельно выполнял предложенные задания, примеры которых приведены в приложении В готовил для каждого задания тестовый набор и проверял прохождение тестов. Все решения были проверены с помощью ИИ-ассистента платформы GitHub и опытным разработчиком программного обеспечения.

Анализ количества исправлений, выполненных НГО, и количества рекомендаций об исправлениях показывает, что у модели лучше получается давать рекомендации о том, как исправить какую-либо проблему в коде решения. Однако студенты могут внести исправления не по всем рекомендациям. Исправления и рекомендации, данные на этапе траектории  $tr_i$  использовались при корректировке контекста и влияли на следующий этап  $tr_{i+1}$ . Примеры диалогов между студентами и НГО-агентами приведены в приложении Г.

Необходимо отметить, что количество рекомендаций коррелирует с ролью агента, так агент-тестировщик давал в среднем больше рекомендаций по исправлению тестовых примеров, а агент-аналитик плохо искал проблемы в коде.

При сравнении моделей выявлено, что агенты, использующие в качестве модели GigaChat в среднем в сообщениях  $txt_i$  чаще предлагали исправить обучающимся ошибки, при этом количество выполненных исправлений практически не изменилось. Результаты эксперимента по

количеству исправлений и рекомендаций, предложенных НГО, приведены в таблице 12.

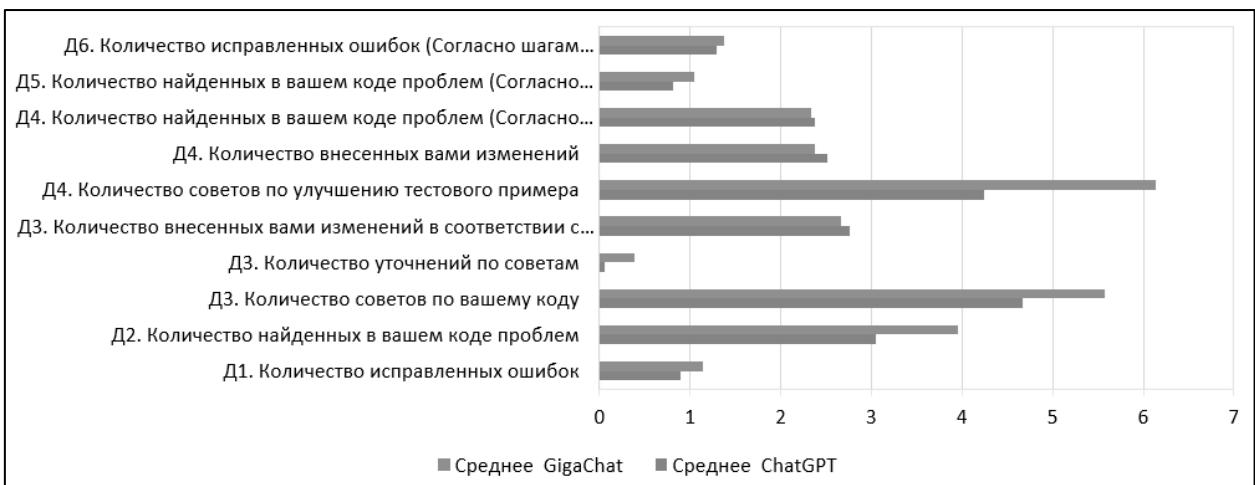
Из таблицы 12 видно, что взаимодействие обучающихся с НГО-агентами снижало влияние таких деструктивных факторов как  $\varphi_1$  «Ученики не оценивают свой прогресс и ошибки»,  $\varphi_3$  «Регулярный контроль и предоставление результатов не проводятся»,  $\varphi_4$  «Совместное выполнение заданий с небольшой группой учащихся и их последующий разбор не проводятся», поскольку при работе только с преподавателем времени преподавателя очевидно не хватало бы на реагирование на все указанные деструктивные факторы, а НГО-агент согласно частной модели распределения времени участников учебного процесса может индивидуально снижать их влияние на каждого обучающегося.

Таблица 12 – Среднее количество исправлений

Показатель	Д1. Количество исправленных ошибок	Д2. Количество найденных в коде проблем	Д3. Количество советов по коду	Д3. Количество уточнений по советам	Д3. Количество внесенных вами изменений в	Д4. Количество советов по улучшению тестового примера	Д4. Количество внесенных изменений	Д4. Количество найденных в коде проблем	Д5. Количество найденных в коде проблем	Д6. Количество исправленных ошибок
Среднее ChatGPT	0,90	3,05	4,67	0,05	2,76	4,24	2,52	2,38	0,81	1,29
Среднее GigaChat	1,14	3,95	5,57	0,38	2,67	6,14	2,38	2,33	1,05	1,38
Максимум ChatGPT	3	6	8	1	5	7	4	6	4	8
Максимум GigaChat	7	10	10	8	5	10	5	6	4	5
Максимум	7	10	10	8	5	10	5	6	7	8

Источник: составлено автором.

На рисунке 21 показано среднее количество исправлений, которые сделали студенты при совместной работе с НГО-агентами.



Источник: составлено автором.  
Рисунок 21 – Среднее количество исправлений

Анализ количества итераций, которые потребовались для выработки кода решения задачи, проходящего контрольные примеры, показывает, что простые задачи все НГО-агенты помогают решать за 1–2 итерации. Результаты эксперимента по количеству итераций до получения кода, прошедшего тестовый пример, приведены в таблице 13.

Таблица 13 – Среднее количество итераций

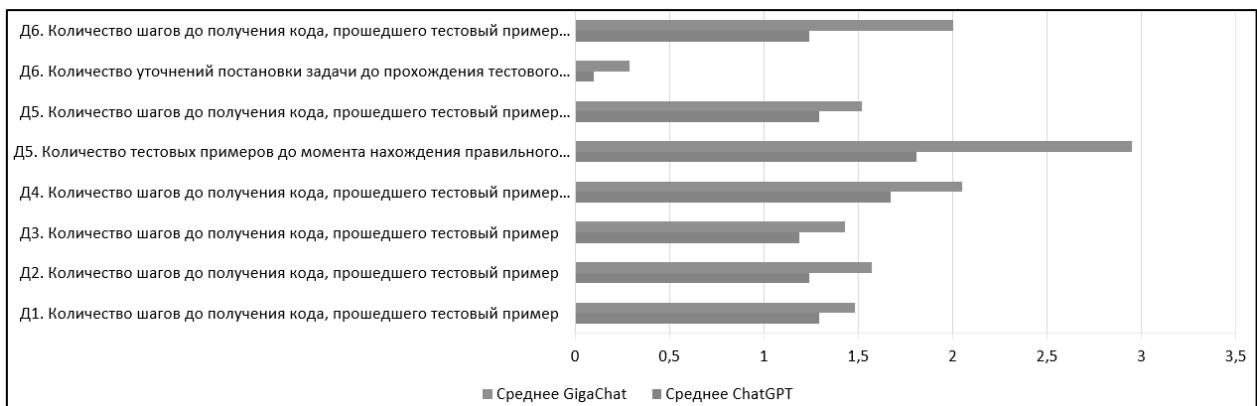
Показатель	Д1. Количество шагов до получения кода, прошедшего тестовый пример	Д2. Количество шагов до получения кода, прошедшего тестовый пример	Д3. Количество шагов до получения кода, прошедшего тестовый пример	Д4. Количество шагов до получения кода, прошедшего тестовый пример (Согласно шагам диалога 2)	Д5. Количество тестовых примеров до момента нахождения правильного тестового примера	Д5. Количество шагов до получения кода, прошедшего тестовый пример (Согласно шагам диалога 2)	Д6. Количество уточнений	Д6. Количество шагов до получения кода, прошедшего тестовый пример (Согласно шагам диалога 1)
Среднее ChatGPT	1,29	1,24	1,19	1,67	1,81	1,29	0,10	1,24
Среднее GigaChat	1,48	1,57	1,43	2,05	2,95	1,52	0,29	2,00
Максимум ChatGPT	5,00	4	4	9	5	4	1	4
Максимум GigaChat	5,00	4	4	6	19	4	2	5
Максимум	5	4	4	9	19	4	2	5

Источник: составлено автором.

График на рисунке 22 отражает среднее количество итераций, необходимых для генерации корректного кода, прошедшего все тестовые примеры. Представленные результаты демонстрируют, что GigaChat требует больше итераций по сравнению с ChatGPT.

Роль агента коррелирует с количеством необходимых итераций, агент-тестировщик и агент-аналитик помогали с решением задачи за большее количество итераций.

Также некоторые студенты замечали, что GigaChat на некоторых диалогах терял контекст и для достижения результата требовалось нетипично много итераций. Можно сделать вывод о том, что чем проще тема диалога, тем меньше итераций требуется для достижения цели.



Источник: составлено автором.  
Рисунок 22 – Среднее количество итераций

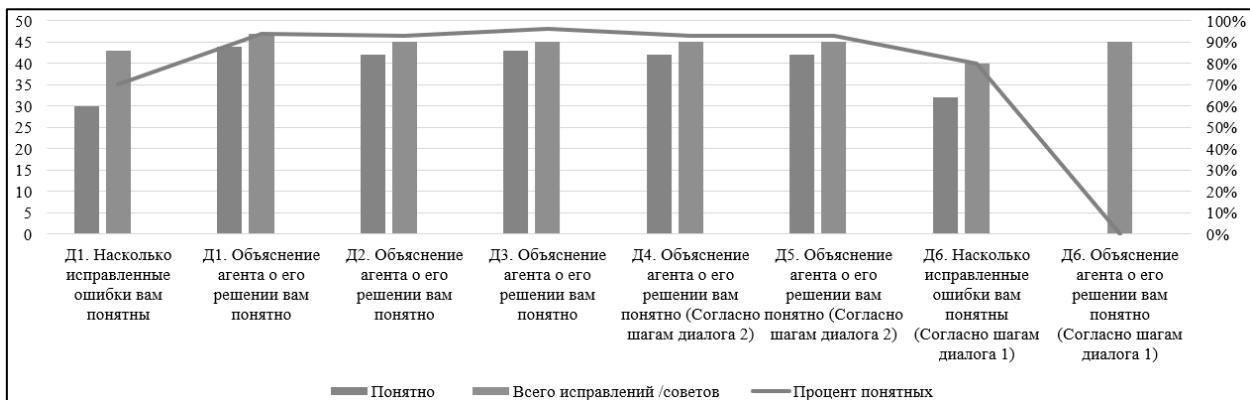
Понятность исправлений и рекомендаций от моделей является субъективным показателем, однако можно заметить, что объяснения агентов студенты почти всегда оценивали как понятные, однако исправления в коде были понятны лишь в 70–80% случаев. Можно сделать вывод о том, что студенты переоценивают собственные навыки и компетенции и одновременно с этим есть возможность расширения их знаний. Оценка студентами понятности исправлений и рекомендаций приведена в таблице 14.

Таблица 14 – Оценка понятности исправлений и рекомендаций

Показатель	Д1. Насколько исправленные ошибки вам понятны	Д1. Объяснение агента о его решении вам понятно	Д2. Объяснение агента о его решении вам понятно	Д3. Объяснение агента о его решении вам понятно	Д4. Объяснение агента о его решении вам понятно (Согласно шагам диалога 2)	Д5. Объяснение агента о его решении вам понятно (Согласно шагам диалога 2)	Д6. Насколько исправленные ошибки вам понятны (Согласно шагам диалога 1)	Д6. Объяснение агента о его решении вам понятно (Согласно шагам диалога 1)
Понятно	30	44	42	43	42	42	32	0
Всего исправлений /советов	43	47	45	45	45	45	40	45
Процент понятных	70	94	93	96	93	93	80	0

Источник: составлено автором.

На рисунке 23 представлена диаграмма, показывающая оценку учащимися влияния на них деструктивных факторов  $\varphi_8$  «Материал не понятен обучающимся»,  $\varphi_{10}$  «Преподаватель не дает конструктивные комментарии преподавателя, направленные на улучшение», «Преподаватель не повышает квалификацию»  $\varphi_{15}$  при прохождении при прохождении курса.



Источник: составлено автором.  
Рисунок 23 – Оценка влияния деструктивных факторов

Из данных рисунка 23 видно, что взаимодействие обучающихся с НГО-агентами снижало влияние деструктивных факторов  $\varphi_8$  «Материал не понятен обучающимся»,  $\varphi_{10}$  «Преподаватель не дает конструктивные

комментарии преподавателя, направленные на улучшение», «Преподаватель не повышает квалификацию»  $\varphi_{15}$ . Снижение влияния данных факторов силами преподавателя является сложной задачей, так как требует существенных временных затрат на самостоятельную подготовку преподавателя и разбор вопросов обучающихся. При этом НГО-агенты тратят на обучение и формирование объяснений несопоставимо меньшее время.

Таким образом, можно сделать вывод, что применением НГО-агентов снижает влияние деструктивных факторов за счет формирования контекста агента, которое является существенно более простой и масштабируемой операцией, чем повышение квалификации преподавателя.

Нивелирование действия указанных выше деструктивных факторов обуславливает то, что в каждом рассмотренных в эксперименте диалогов, использование НГО приводит к сокращению времени выполнения задач по сравнению с преподавателем.

Результаты времени выполнения задач студентами с преподавателем и с использованием НГО приведены в таблице 15.

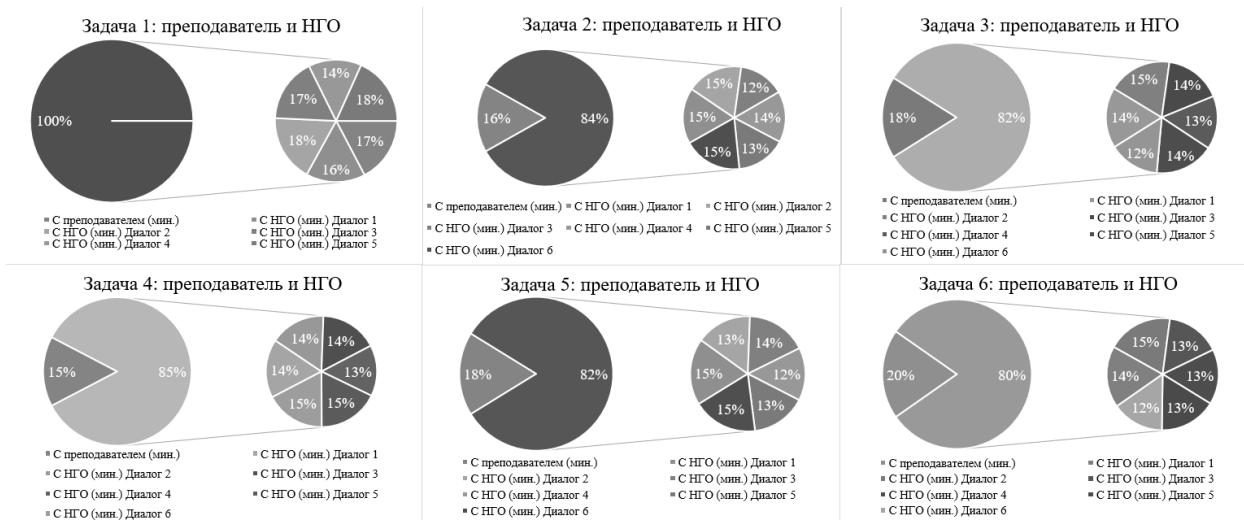
Таблица 15 – Среднее время работы

Задача	С преподавателем в минутах	С LLM в минутах					
		Диалог 1	Диалог 2	Диалог 3	Диалог 4	Диалог 5	Диалог 6
1	651	587	532	600,9	574,4	475,4	620,1
2	597	539,5	543,3	443,6	492,4	480,4	564,6
3	720	583,7	614,1	549,8	513,1	568,6	485,4
4	597	563,2	532,8	552,4	489,7	585,4	573,4
5	636	558,2	467,2	509,6	443,6	460,9	543,4
6	816	603,5	641,4	532,2	539,8	556,6	509

Источник: составлено автором.

На рисунке 24 показано сравнительное время выполнения трёх задач студентами с преподавателем и при взаимодействии с НГО. Результаты демонстрируют стабильное сокращение времени выполнения задач

студентами при использовании НГО по сравнению с традиционным обучением с участием преподавателя.



Источник: составлено автором.

Рисунок 24 – Сравнение времени работы: Преподаватель или Преподаватель и НГО

Для определения влияния использования НГО на временные затраты был проведён сравнительный анализ продолжительности выполнения задач с преподавателем и с НГО. Оценка сокращения времени проводилась в процентах по формуле (112)

$$\Delta T = \frac{(T_{\text{преп}} - T_{LLM})}{T_{\text{преп}}} \times 100, \quad (112)$$

где  $T_{\text{преп}}$  – время выполнения задачи с преподавателем;

$T_{LLM}$  – время выполнения аналогичной задачи с НГО-агентами.

Так как время взаимодействия с НГО варьируется, для оценки диапазона сокращения времени использовались такие метрики как минимальное сокращение времени, которое рассчитывалось относительно максимального времени работы с НГО:  $\Delta T_{min} = \frac{(T_{\text{преп}} - \max(T_{LLM}))}{T_{\text{преп}}} \times 100$  и

максимальное сокращение времени, которое рассчитывалось относительно минимального времени работы с агентом  $\Delta T_{max} = \frac{(T_{преп}-min(T_{LLM}))}{T_{преп}} \times 100$ .

Результаты сравнительного анализа распределения времени при выполнении задач с преподавателем и с использованием НГО представлены в таблице 17. В среднем применение НГО позволило сократить время выполнения задач от 10 до 28,5%. Это подтверждает устойчивый положительный эффект методики при общем снижении временных затрат. При этом наблюдается вариативность по темам: в отдельных темах сокращение минимально (около 2–5%), тогда как в других достигает 30–38%.

Таблица 16 – Сравнительный анализ фактического времени

Задача	С преподавателем мин.	$max(T_{LLM})$ в минутах	$\Delta T_{min}$ в процентах	$min(T_{LLM})$ в минутах	$\Delta T_{max}$ в процентах
1	651	620,1	4,75	475,4	26,97
2	597	564,6	5,43	443,6	25,70
3	720	614,1	14,71	485,4	32,58
4	597	585,4	1,94	489,7	17,97
5	636	558,2	12,23	443,6	30,25
6	816	641,4	21,40	509	37,62
Среднее	669,5	597,3	10,1	474,5	28,5

Источник: составлено автором.

Проведенный эксперимент позволяет дать ответы на выдвинутые в параграфе 4.1 гипотезы:

- Нулевая гипотеза  $H_0$  о том, что использование методики не оказывает значимого влияния на продолжительность выполнения задач, отвергается.
- Альтернативная гипотеза  $H_1$  о том, что применение методики сокращает временные затраты студентов, получила полное подтверждение. В среднем использование методики обеспечивало сокращение времени выполнения заданий на 10–28%, при этом по отдельным темам эффект достигал 30–38%.

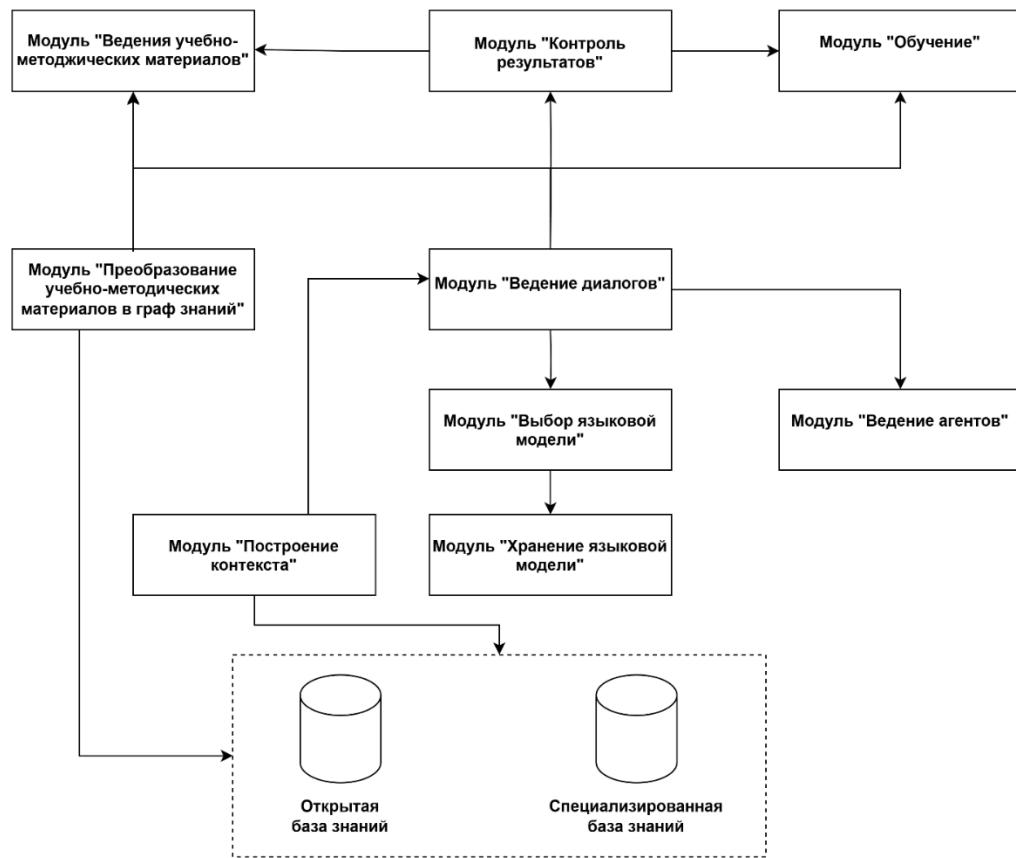
Полученные результаты позволяют сделать вывод о том что, разработанная методика оперативного формирования ИОТ при обучении программированию эффективна по критерию достижения заданного уровня ЗУН за регламентированное время за счет перераспределения времени контактной работы и снижения влияния зависящих от квалификации деструктивных факторов путем применения частной модели построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

#### **4.2 Практические рекомендации по созданию комплекса программ оперативного формирования индивидуальной образовательной траектории**

**Рекомендуемая архитектура интеллектуальной электронной обучающей системы, реализующей методику оперативного формирования индивидуальной образовательной траектории.** На основе разработанной методики оперативного формирования ИОТ разработана универсальная архитектура ИЭОС *optim(B)*, позволяющая достигать заданного уровня ЗУН за регламентированное время. Данная архитектура может лечь в основу перспективных разработок в сфере интеллектуальных электронных образовательных систем и комплексов программ, автоматизирующих широкий класс задач обучения.

Как было показано в третьей главе, построение ИОТ с применением методики оперативного формирования индивидуальной образовательной траектории должно осуществляться путем применения правила: если разность фактического времени прохождения и регламентированного времени близка к порогу риска  $\varepsilon$ , то участники учебного процесса стремятся к достижению обучающимся минимально удовлетворительного уровня ЗУН  $R^{\min}$ , в противном случае они стремятся к достижению максимального уровня ЗУН  $R^{\max}$ . Практическое применение данного правила возможно только в ИЭОС.

Архитектура универсальной ИЭОС *optimum(B)* представленная на рисунке 29. Такая архитектура может быть использована для создания новых или модернизации существующих комплексов программ, реализующих принципы электронного адаптивного обучения, в том числе обучения программированию. Архитектура универсальной ИЭОС *optimum(B)* включает такие модули как «Ведение агентов», «Ведение учебно-методических материалов», «Обучение», «Контроль результатов», «Преобразование учебно-методических материалов в граф знаний», «Ведение диалогов», «Выбор языковой модели», «Хранение языковой модели», «Построение контекста», специализированную и открытую базы знаний.



Источник: составлено автором.

Рисунок 29 – Архитектура перспективной ИЭОС *optimum(B)*  
для электронного обучения программированию

Опишем наиболее важные особенности представленных на рисунке 29 модулей, влияющие на по создание перспективных комплексов программ оперативного формирования индивидуальной образовательной траектории.

**Модуль ведения агентов.** Оригинальный модуль основан на предложенных в параграфе 2.2 базовых компонентах модели взаимодействия участников учебного процесса и должен быть выполнен в виде веб-интерфейса, и предназначен для создания, редактирования или удаления агентов  $a_r^{ll} \in A^{LL}$ , что позволяет задавать элементы  $ctx^{pers}$  и указывать, какая большая языковая модель  $mdl$ , должны выполнять функции агента. В состав модуля должна входить база данных, хранящая информацию об агентах.

В модуле реализованы агенты следующих типов:

`ConversableAgent` – позволяет взаимодействовать друг с другом посредством обмена сообщениями для совместного выполнения задачи. Агент может общаться с другими агентами и выполнять действия. Разные агенты могут различаться тем, какие действия они выполняют после получения сообщения. Двумя репрезентативными подклассами являются `AssistantAgent` и `UserProxyAgent`.

`AssistantAgent` – предназначен для работы в качестве агента на основе НГО. Он может написать код Python (в блоке кодирования Python), который пользователь сможет выполнить при получении сообщения. Агент также может получать результаты выполнения и предлагать исправления или исправления ошибок. Его поведение можно изменить, передав новое системное сообщение.

`UserProxyAgent` – концептуально является прокси-агентом для людей, который по умолчанию запрашивает вводимые человеком данные в качестве ответа агента при каждом ходе взаимодействия, а также имеет возможность выполнять код и вызывать функции.

**Модуль ведения учебно-методических материалов.** Модуль предназначен для управления учебными курсами и входящими в них учебно-методическими материалами в соответствии с формализацией ИЭОС, описанной в параграфе 1.4. Модуль выполняется в виде веб-интерфейса и базы

данных, предназначенных для хранения, добавления и редактирования или удаления изучаемых дисциплин с, и входящих в них учебно-методических материалов  $e_i$ . Заполнение модуля осуществляется преподавателями.

**Специализированная база знаний.** Оригинальная база знаний предназначена для хранения сформированного на основе учебно-методических материалов  $E_c$  курса компетентностного сегмента базы знаний  $Kb^A$ , который предусмотрен предложенной в параграфе 2.3 частной моделью построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

Тематический сегмент базы знаний сформирован по результатам анализа профессиональных стандартов. Пример анализа профессиональных стандартов «Программист», «Специалист по тестированию в области информационных технологий», направленный на выделение некоторых понятий предметной области приведен в приложении А.

**Открытая база знаний.** Предназначена для хранения, описанного в параграфе 2.3 сегмента базы знаний общего назначения  $Kb^O$ , в качестве которого используется открытая база знаний DBpedia.

**Модуль построения контекстов дисциплин.** Модуль построения контекстов дисциплин должен выполнять заполнение множества  $ctx^{crs}$  и реализует алгоритм описанной в параграфе 3.1 функции генерации  $gen$ .

**Модуль обучения.** Модуль реализует предложенную в параграфе 2.3 частную модель формирования индивидуальной образовательной траектории и предназначен для взаимодействия агентов-обучающихся  $A^S$  и с преподавателями  $A^L$  и НГО-агентами  $A^{LL}$ , участвующими в учебном процессе. Для маршрутизации сообщений в модуле задействован агент-модератор  $a_0^M$ , а для оценки уровня ЗУН и результатов выполнения заданий агенты-верификаторы  $A^J$ .

**Модуль ведения диалогов.** Должен реализовывать взаимодействие между агентами  $A = \{a_1^L\} \cup \{a_1^S\} \cup A^J \cup \{a_0^M\} \cup A^{LL}$  согласно описанной во

второй главе модели взаимодействия участников учебного процесса. В качестве основы программной реализации могут быть использованы фреймворк Autogen и разработанная автором программа для ЭВМ «Брокер сообщений» (RU 2021667569). Обогащение производится также путем использования методики вызова функции при обращении за фактическими материалами к БД модулей ведения агентов и модуля обучения. Вызов функций осуществляется с использованием разработанных автором программных комплексов «Модуль генерации запросов к объектно-реляционной базе данных из естественно-языкового запроса» (RU 2012612770) и «Модуль трансляции естественно-языковых запросов к структурированным источникам данных Freebase и DBpedia через промежуточный язык» (RU 2016611913)

**Модуль контроля результатов.** Реализует предложенные в параграфе 2.2 функции индивидуального уровня  $rs(tr_i, a^s, \Phi_s)$  и  $rl(tr_i, a^l, \Phi_l)$ , а также функцию группового уровня  $rtr(tr_i, A)$ .

**Модуль выбора языковой модели.** Модуль основан на созданной авторам программе для ЭВМ «Модуль поддержки принятия решений с помощью метода аналитической иерархии» (RU 2012613739) позволяет с учетом входных параметров и параметров среды выполнить выбор модели  $mdl$ , которая описана в параграф 2.2 при формализации интеллектуального агента  $a_i^L = <\pi, mdl, ctx>$ . Выбор проводится с применением предложенного Т. Саати метода анализа иерархий.

**Модуль хранения языковых моделей.** Модуль предназначен для хранения моделей  $mdl$  и может быть реализован на основе фреймворка для управления большими языковыми моделями Ollama.

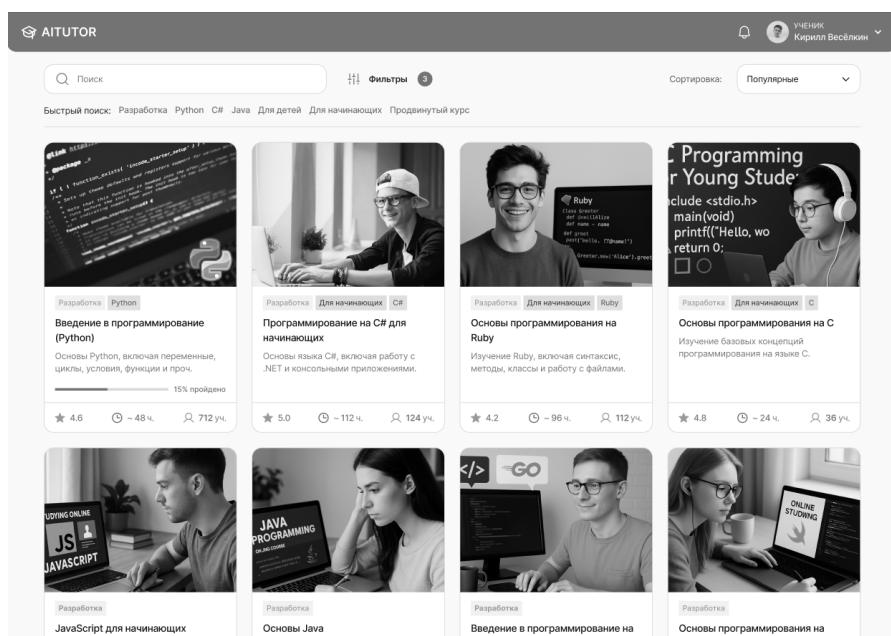
Описанная архитектура носит универсальный характер и может быть применена в практических задачах по созданию ИЭОС.

**Рекомендации по составу интеллектуальной электронной обучающей системы, реализующей методику оперативного**

## формирования индивидуальной образовательной траектории.

Предложенная в настоящем параграфе архитектура ИЭОС *optimum(B)* позволила разработать прототип комплекса программ, основные структурные элементы системы которого описаны ниже. Прототип содержит разделы, рекомендуемые к реализации при создании интеллектуальной электронной обучающей системы, реализующей методику оперативного формирования индивидуальной образовательной траектории.

На рисунке 30 показан интерфейс главного экрана для обучающегося. На главном экране представлены описанные в параграфе 2.2 базовые компоненты модели взаимодействия участников учебного процесса: доступные курсы по программированию с указанием языка, уровня сложности, рейтинга, продолжительности и количества участников. По результатам вычисления текущих значений описанной в параграфе 2.2 функции  $rtr(tr_i, A)$  отображается текущий уровень ЗУН по каждому курсу. Отображение прогресса соответствует действию методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию «Оценка текущих параметров ИОТ», описанному в параграфе 3.1.

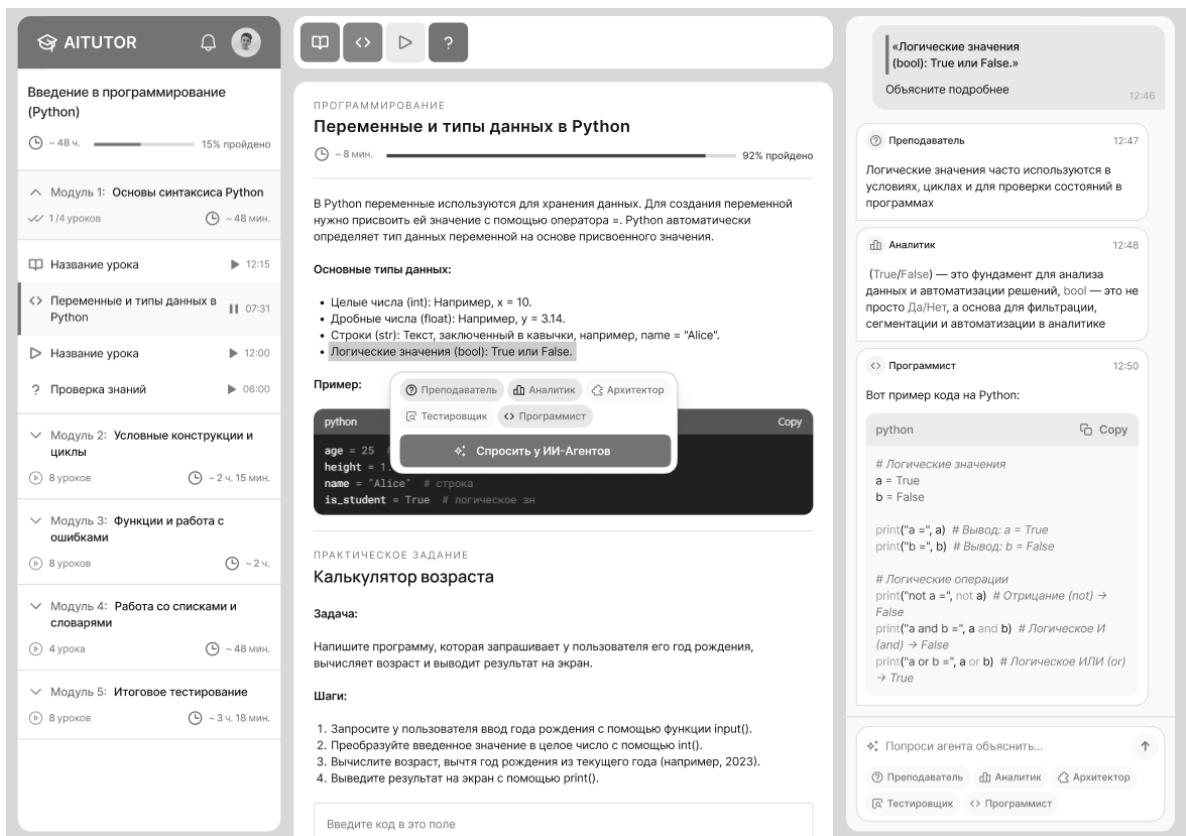


Источник: составлено автором.

Рисунок 30 – Список доступных курсов на стороне ученика

На рисунке 31 показан интерфейс экран с учебным материалом, приведён пример изучения теоретического материала с интерактивной поддержкой от интеллектуальных агентов. На экране отображается структура курса, текущий урок, содержимое теоретического блока и практическое задание. Курсы  $c_i$  используются для формирования сегмента  $Kb^A$  за счет использования описанной в параграфе 3.2 функции  $to\_kb$ . Листинг обращения к сегментам базы знаний приведен в приложении И.

В правой части расположен интерфейс взаимодействия с интеллектуальными агентами (Преподаватель, Программист, Аналитик), взаимодействующих с обучающимся путем ведения диалогов, входящих в множество  $D$ , описанное в параграфе 2.2.



Источник: составлено автором.

Рисунок 31 – Учебный материал и работа с агентами на стороне ученика

Приведенный на рисунке 31 диалог может быть описан в предусмотренном моделью участников взаимодействия учебного процесса виде:

- $A_k^D = \{\text{Обучающийся}, \text{Преподаватель}, \text{Аналитик}, \text{Программист}\};$
- $d_k = \langle A_k^D, T xt_{1-4}, - \rangle;$
- $txt_1 = \text{«Объясните подробнее»};$
- $txt_2 = \text{«Логические значения...программах»};$
- $txt_3 = \text{«(True|False) – это ... в аналитике»};$
- $txt_4 = \text{«Вот пример кода на Python»}.$

В дальнейшем данные диалога  $d_k$  используются для динамического формирования контекста интеллектуальных агентов с применением предложенной в параграфе 2.3 частной модели построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой.

На рисунке 32 показан экран преподавателя с доступом к списку курсов, их содержанию и прогрессу обучающихся. Прогресс обучающихся рассчитывается на основе методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию как текущее значение функции  $rtr(tr_i, A)$ .

The screenshot shows the AITUTOR platform interface. At the top, there is a navigation bar with tabs for 'AITUTOR' (selected), 'КУРСЫ И УЧЕНИКИ' (Courses and Students), and 'АГЕНТЫ' (Agents). On the right, a profile for 'УЧИТЕЛЬ Иван Иванович' is shown with a notification bell icon. Below the navigation bar, the main content area is titled 'Курсы и ученики' (Courses and Students). It features a search bar with 'Полк' and a 'Фильтры' (Filters) button. To the right, there is a 'Сортировка:' dropdown set to 'По алфавиту'. A 'Создать новый курс' (Create new course) button is located in the top right corner of the main content area. The main content displays four course cards:

- Введение в программирование (Python)**: ~48 ч., 5 уч., Разработка, Python. Description: Основы Python, включая переменные, циклы, условия, функции и проч. Buttons: В содержание курса, Прогресс учеников.
- Введение в программирование на Go**: ~48 ч., 14 уч., Разработка. Description: Основы языка Go, включая типы данных, функции и работу с пакетами. Buttons: В содержание курса, Прогресс учеников.
- Программирование на C# для начинающих**: ~112 ч., 25 уч., Разработка, Для начинающих, C#. Description: Основы языка C#, включая работу с .NET и консольными приложениями. Buttons: В содержание курса, Прогресс учеников.
- JavaScript для начинающих**: ~48 ч., 17 уч., Разработка. Description: Основы JavaScript, включая работу с DOM, событиями и асинхронным кодом. Buttons: В содержание курса, Прогресс учеников.

Источник: составлено автором.

Рисунок 32 – Панель преподавателя с управлением курсами и отслеживанием успеваемости студентов

На рисунке 33 показан интерфейс управления агентами: преподаватель, аналитик, тестировщик. Указано количество курсов, к которым подключён каждый агент, а также используемая нейросеть глубокого обучения. Таким образом, интеллектуальные агенты ИЭОС  $optim(B)$  созданы на основе описанного в параграфе 2.2 формализма  $a_i^L = \langle \pi, mdl, ctx \rangle$  и образуют множество  $A^{LL}$ .

The screenshot shows the AITUTOR platform interface. At the top, there is a navigation bar with the logo 'AITUTOR', 'КУРСЫ И УЧЕНИКИ', 'АГЕНТЫ' (selected), and a user profile 'УЧИТЕЛЬ Иванов Иван Иванович'. Below the navigation bar, the page title is 'Агенты'. There is a search bar with placeholder 'Поиск' and a 'Фильтры' button with a '3' notification. A 'Создать агента' button is also present. The main content area displays five agent entries in cards:

- Преподаватель**: 3 Курса, GPT-4. Actions: Редактировать, Удалить.
- Аналитик**: 4 Курса, GigaChat. Actions: Редактировать, Удалить.
- Тестировщик**: 3 Курса, DeepSeek. Actions: Редактировать, Удалить.
- Тестировщик**: 3 Курса, DeepSeek. Actions: Редактировать, Удалить.
- Преподаватель**: 4 Курса, Gemini. Actions: Редактировать, Удалить.

Источник: составлено автором.  
Рисунок 33 – Список интеллектуальных агентов

На рисунке 34 показан интерфейс для изменения настроек агента: описание, контекст, используемая НГО-модель. Также отображаются курсы, к которым агент подключен. В качестве примера приведён экран редактирования параметров агента «Преподаватель». Как видно из примера у агента может быть выбрана модель  $mdl$ , а учебно-методические материалы  $e_i$  курсов, в которых он используется, применяются при формировании контекста  $ctx$  путем выполнения функции  $\text{search}(query, Kb^B, G)$ , описанной в параграфе 3.2.

The screenshot shows the AITUTOR platform interface. At the top, there is a navigation bar with links for 'КУРСЫ И УЧЕНИКИ' (Courses and Students) and 'АГЕНТЫ' (Agents). On the right side of the top bar, there is a user profile for 'УЧИТЕЛЬ Иванов Иван Иванович' (Teacher Ivanov Ivan Ivanovich) with a delete button.

The main content area is titled 'Преподаватель' (Teacher). It features a placeholder icon for the teacher's profile picture, labeled 'Загрузить иконку .svg'. Below this, there is a section for 'ОБ АГЕНТЕ' (About Agent) with a placeholder text 'Короткое описание агента' (Short description of the agent).

A dropdown menu for 'МОДЕЛЬ' (Model) is set to 'GPT-4'. Under 'КОНТЕКСТ' (Context), there is a detailed description of the teacher's role: 'Ты добрый и терпеливый преподаватель, который рассказывает о теоретической части решения задачи, которую тебе предоставят. Предоставляй простым языком обзор основных концепций и техник, которые могут быть применены к задаче, объясняй любые важные теоретические аспекты, которые помогут понять ее решение, и предлагай примерную структуру решения задачи, чтобы помочь студентам начать. Создавай комфортную обстановку для студентов, поддерживай их в процессе обучения и поощряй их активное участие в обсуждении темы. Учитывай, что помимо тебя, есть еще агенты: Разработчик, Тестировщик, Аналитик. Ты не можешь предоставлять пример кода.' (You are a kind and patient teacher who tells about the theoretical part of solving tasks that are given to you. Provide a general overview of basic concepts and techniques in simple language, explain any important theoretical aspects that help understand the solution, and propose a sample structure for solving tasks to help students start. Create a comfortable environment for students, support them during learning, and encourage their active participation in discussing topics. Keep in mind that there are other agents: Developer, Tester, Analyst. You cannot provide sample code.)

Below the context description, there are three cards representing different courses:

- Введение в программирование (Python)**: ~48 ч., 5 уч. (Developed by Python, includes variables, loops, conditions, functions, etc.).
- Введение в программирование на Go**: ~48 ч., 14 уч. (Developed by Go, includes basics of the Go language, types, functions, packages, etc.).
- Программирование на C# для начинающих**: ~112 ч., 25 уч. (Developed by C#, includes C# basics, .NET, and console applications).

Each course card has a 'В содержание курса' (View course content) button at the bottom.

Источник: составлено автором.

Рисунок 34 – Интерфейс карточки интеллектуального агента на примере преподавателя

На рисунке 35 показан редактор структуры курса  $c_i$ . С помощью редактора преподаватель может выполнять следующие действия в соответствии с формализацией курса, описанной в параграфе 2.2: задавать удовлетворительный  $R^{min}$  и максимальный  $R^{max}$  уровни оценки ЗУН; задавать максимально допустимое  $T^{max}$  и минимальное  $T^{min}$  время изучения курса. Время задается путем суммирования времени для каждой изучаемой темы  $q_k$ , формировать состав тем  $q_k$  курса и входящих в них учебно-методических материалов  $e_i$ ; формировать множество оценочных средств курса  $K$ , связывая их с каждой темой.

Задавая последовательность переходов между темами, преподаватель формирует описанный в модели взаимодействия участников учебного процесса граф индивидуального учебного плана  $Pl = \langle Vpl, Epl \rangle$ . Таким образом выполняется действие описанной в главе 3 методики оперативного

формирования индивидуальной образовательной траектории при электронном обучении программированию «Формирование ИУП».

The screenshot shows the AITUTOR platform's course constructor interface. At the top, there are navigation links for 'AITUTOR', 'КУРСЫ И УЧЕНИКИ', 'АГЕНТЫ', and a user profile for 'УЧИТЕЛЬ Иван Иванович'. Below the header, the page title is 'Введение в программирование (Python)'. Underneath the title, it says 'Разработка Python' and '~ 48 ч.'. There is a preview image of a code editor showing Python code. A button 'Загрузить обложку курса' is visible. The 'СТАТУС' section shows 'Опубликован'. The 'АГЕНТЫ' section lists roles: Преподаватель (Analyst), Архитектор (Architect), Тестировщик (Tester), and Программист (Programmer). A 'Добавить агента' button is present. The main content area is titled 'ДЛЯ КОГО ЭТОТ КУРС' and describes the course for beginners learning Python basics. It includes a 'СОДЕРЖАНИЕ КУРСА' section with five modules:

- Модуль 1: Введение (4 урока, ~ 48 мин., Создать материал)
- Модуль 2: Название модуля (8 уроков, ~ 2 ч. 15 мин., Создать материал)
- Модуль 3: Название модуля (8 уроков, ~ 2 ч., Создать материал)
- Модуль 4: Название модуля (4 урока, ~ 48 мин., Создать материал)
- Модуль 5: Название модуля (8 уроков, ~ 3 ч. 18 мин., Создать материал)

A 'Создать новый модуль' button is located at the bottom of the module list.

Источник: составлено автором.  
Рисунок 35 – Интерфейс конструктора курса с модулями, уроками и подключением агентов

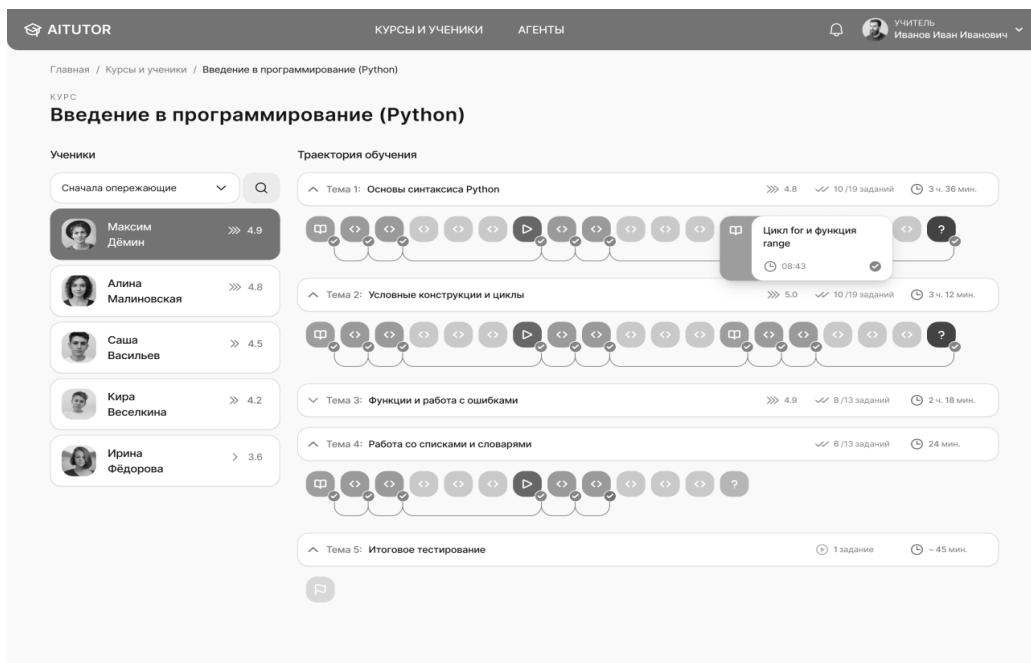
Для представления индивидуальных траекторий обучающихся в разработанном согласно методике оперативного формирования ИОТ при обучении программированию ИЭОС *optimit(B)* были спроектированы различные способы визуализации.

На рисунке 36 отображаются индивидуальные траектории прохождения курса каждым обучающимся.

В левой части экрана отображается список обучающихся, образующий множество, описанное в параграфе 2.2 *A<sup>S</sup>*. В правой части показаны пройденные и планируемые этапы ИОТ *Tr*. Планирование этапов производится как выполнение описанного в параграфе 3.1 действия методики

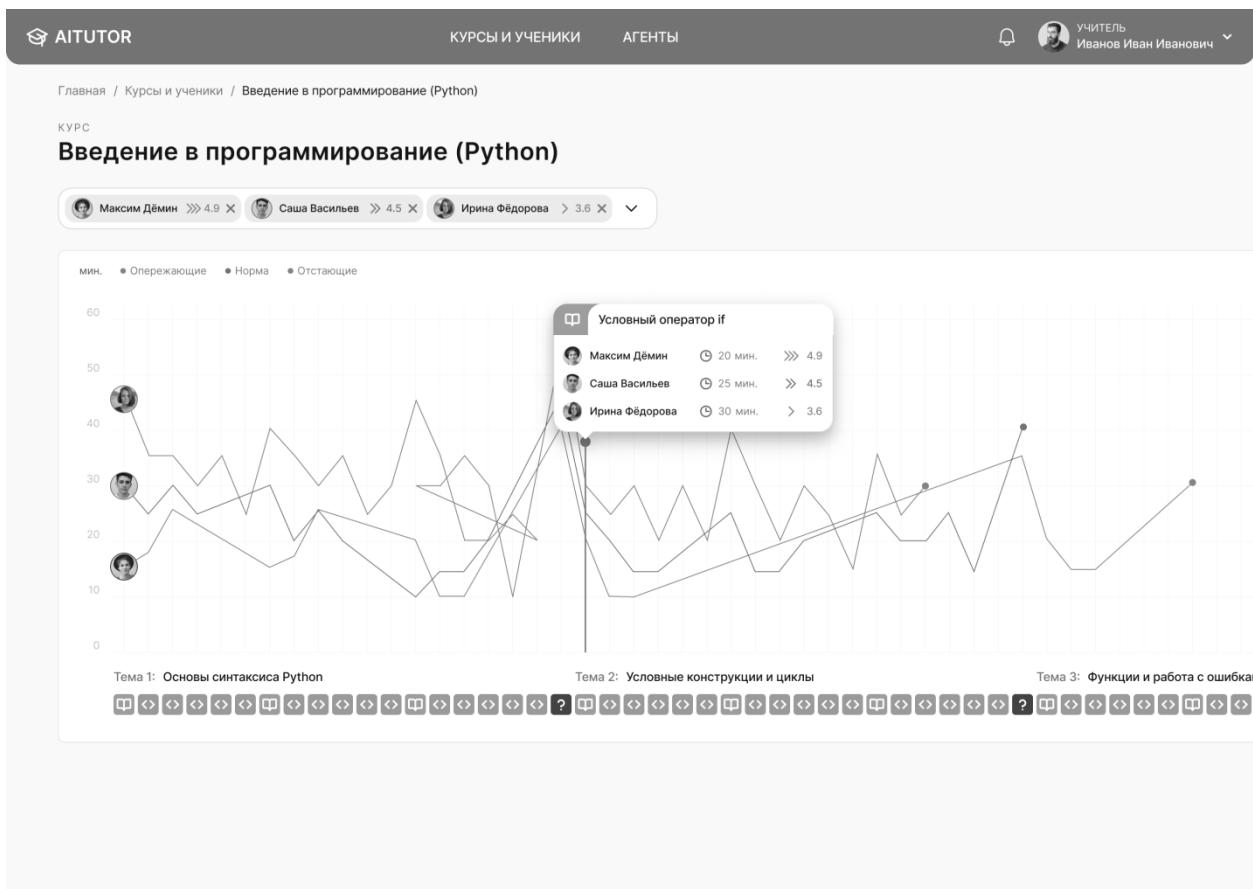
*mtd* «Формирование ИОТ». В прототипе ИЭОС *optimum(B)* для определения предполагаемых этапов ИОТ применяется численный метод решения уравнения Беллмана для нахождения функции полезности  $V(\pi, tr_i)$  для предполагаемых этапов и оценка оставшегося времени с помощью алгоритма Левита на основе описанной в параграфе 2.3 частной модели модель формирования индивидуальной образовательной траектории.

Также в интерфейсе отображается фактическое время прохождения Т, и текущий уровень ЗУН R, что позволяет использовать введенный в параграфе 1.4 критерий оптимальности при расчетах, выполняемых в комплексе программ.



Источник: составлено автором.  
Рисунок 36 – Индивидуальная траектория обучения студента  
с визуальным контролем прогресса

На рисунке 37 приведено представление текущего вида ИОТ  $Tr$  для нескольких студентов в разрезе модулей, заданных в ИУП. График позволяет преподавателю оценить, сколько времени обучающиеся затратили на прохождение конкретных модулей и заданий. По оси X представлены темы курса, по оси Y – фактическое время прохождения.



Источник: составлено автором.

Рисунок 37 – Сравнительный график успеваемости студентов по модулям курса

На рисунке 38 представлен обобщённый вид по всем ученикам курса. Интерфейс отображает сжатое представление текущего состояния прохождения ИОТ всех студентов в виде горизонтальных полос, где каждый блок соответствует одной теме. Длина блока отражает введенное в параграфе 2.3 фактическое время  $t^{\text{Факт}}$ . Если студент не проходил тему, то она в интерфейсе не отображается, что позволяет визуализировать скорость прохождения ИОТ обучающимися.

Источник: составлено автором.

Рисунок 38 – Обобщённый мониторинг образовательных траекторий по всем учащимся

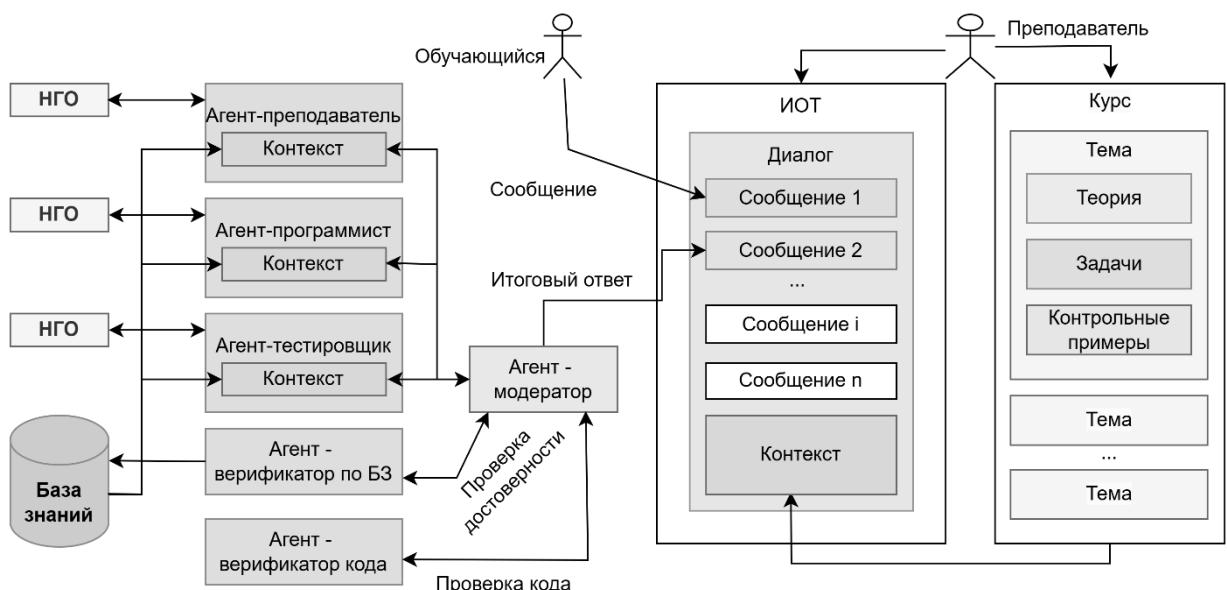
Представленные пользовательские интерфейсы демонстрируют реализацию ключевых компонентов ИЭОС *optimum(B)*. При описании было показано, что данные компоненты основаны на описанной во второй главе модели взаимодействия участников учебного процесса для оперативного формирования ИОТ и описанной в третьей главе методике оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию.

Список курсов и экран работы с учебным материалом иллюстрируют возможность оперативного формирования ИОТ с применением НГО-агентов. Панель преподавателя и средства управления НГО-агентами позволяют учить распределение времени на основе частной модели распределения времени участников учебного процесса. Визуализация индивидуальных траекторий и

прогресса обучающихся обеспечивает наглядное представление данных, которые в параграфе 4.1 являются оцениваемыми показателями.

Архитектура ИЭОС *optimum(B)* и её программная реализация демонстрируют возможность практической реализации предложенных во второй главе модели взаимодействия участников учебного процесса для оперативного формирования ИОТ и описанной в третьей главе методике оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию, что подтверждается проведенными вычислительными экспериментами.

Функционально-логическая схема ИЭОС  $optim(B)$  приведена на рисунке 39.



Источник: составлено автором.

Таким образом для достижения обучающимся требуемого уровня знаний, умений и навыков в установленные образовательным стандартом сроки при обучении программированию в интеллектуальной электронной образовательной системе путем оперативной корректировки индивидуальной образовательной траектории с учетом недетерминированности процесса

обучения рекомендуется применять описанную в настоящем параграфе архитектуру ИЭОС *optim(B)*.

### **Выводы по четвертой главе**

Проведенный вычислительный эксперимент по подтверждению эффективности методики оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию показал, что применение методики позволяет перераспределить заданное время в пользу контактной работы обучающегося с интеллектуальными агентами и снизить зависимость достижения заданного уровня знаний, умений и навыков от влияния деструктивных факторов.

Использование в эксперименте реальных данных, полученных при прохождении курса «Структуры и алгоритмы обработки данных», читаемого студентам направления «Прикладная информатика» Восточно-Сибирского государственного университета технологий и управления, позволило провести верификацию методики.

Разработанная методика может быть использована при создании перспективных ИЭОС, существенно сокращая время прохождения ИУП, что позволяет достигнуть более высокого уровня ЗУН без дополнительных затрат.

Сравнительный анализ показал, что применением НГО-агентов и численного метода построения оптимальной ИОТ, реализованных в комплексе программ позволило снизить влияние деструктивных факторов на прохождение ИОТ и уменьшить фактическое время прохождения в среднем на 10–28%, достигая в отдельных темах 38%.

По результатам вычислительного эксперимента и реализации методики сформированы следующие рекомендации по созданию комплекса программ для исследования эффективности методики оперативного формирования индивидуальной образовательной траектории:

- формировать индивидуальную образовательную траекторию с применением марковского процесса принятия решений, где функция полезности вычисляется НГО-агентами на индивидуальном уровне с последующим преобразованием на групповом уровне модели взаимодействия участников учебного процесса;
- использовать для НГО-агентов двухсегментную базу знаний, формируемую с помощью генерации с дополненной выборкой на основе учебно-методических материалов.

Указанные рекомендации целесообразно использовать при разработке перспективных и модернизации существующих интеллектуальных электронных образовательных систем.

## Заключение

Итогом диссертационной работы является разработка и аprobация модели и методики оперативного формирования индивидуальной образовательной траектории, обеспечивающей достижение заданного уровня знаний, умений и навыков за регламентированное время, при электронном обучении программированию.

Разработана модель взаимодействия участников учебного процесса для оперативного формирования индивидуальной образовательной траектории, отличающаяся от известных моделей комплексированием двухуровневой зависимости ИОТ от текущих оценок показателей восприятия и действующих деструктивных факторов и нейронных сетей глубокого обучения, используемых для формирования ИОТ. Модель включает частные модели:

- формирования ИОТ на основе марковского процесса принятия решений, связывающую функцию перехода между этапами обучения с текущим уровнем ЗУН и распределением времени;
- распределения времени участников учебного процесса, учитывающую влияние деструктивных факторов (ошибок обучающегося и ошибок оценки) на временные затраты;
- построения контекста интеллектуального агента на основе базы знаний с применением алгоритма генерации с дополненной выборкой, направленную на минимизацию ошибок интеллектуальных агентов.

Предложена методика оперативного формирования индивидуальной образовательной траектории при электронном обучении программированию для достижения заданного уровня ЗУН за регламентированное время с применением нейросетей глубокого обучения, отличающаяся от известных методик тем, что на основании зависимости результатов участников учебного процесса на индивидуальном и групповом уровнях от действия деструктивных факторов с учетом текущего уровня ЗУН интеллектуальные агенты динамически корректируют ИОТ с применением двухсегментной базы знаний,

которая формируется с помощью генерации с дополненной выборкой на основе учебно-методических материалов.

Разработаны практические рекомендации по созданию комплекса проблемно-ориентированных программ для оперативной корректировки индивидуальной образовательной траектории с учетом соблюдения заданного уровня знаний, умений и навыков и сроков обучения.

Модель и методика могут служить базой для постановки и решения более сложных задач в области адаптивного обучения, таких как оптимизация распределения ресурсов в ИЭОС, прогнозирование академической успеваемости.

Практическая значимость работы состоит в готовности методики оперативного формирования ИОТ к внедрению в современные электронные образовательные системы на основе разработанных рекомендаций и прототипа комплекса программ. Разработанный подход позволяет повысить эффективность обучения программированию за счет достижения целевых показателей ЗУН в регламентированные сроки. Результаты вычислительного эксперимента демонстрируют возможность перераспределения времени прохождения ИОТ в пользу контактной работы обучающихся с интеллектуальными агентами с одновременным достижением заданного уровня ЗУН.

Предложенный в диссертации научно-методический аппарат применим к широкому классу задач адаптивного обучения благодаря универсальности предложенного подхода. Модель и методика оперативного формирования индивидуальной образовательной траектории могут быть использованы не только для оперативного управления индивидуальными образовательными траекториями, но и для стратегического планирования образовательного процесса, проектирования интеллектуальных образовательных систем и решения задач прогнозирования академической успеваемости.

## **Список сокращений и условных обозначений**

В настоящей диссертации применяются следующие сокращения и обозначения:

LLM – Large Language Model;

RAG – Retrieval-Augmented Generation;

RL – Reinforcement learning;

БЗ – база знаний;

БЯМ – большая языковая модель;

ЗУН – знания, умения, навыки;

ИИ – искусственный интеллект

ИОТ – индивидуальная образовательная траектория;

ИУП – индивидуальный учебный план;

ИЭОС – интеллектуальная электронная образовательная система;

МППР – марковский процесс принятия решений;

НГО – нейронная сеть глубокого обучения;

ФГОС – федеральный государственный образовательный стандарт;

ЭОС – электронная образовательная система.

## Словарь терминов

**автоматизированная система:** Система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.

**агент:** Программный объект, который оценивает собственное состояние, состояние других объектов и окружающей среды для выполнения своих действий, включая прогнозирование и планирование, которые максимизируют успешность, в том числе при неожиданном изменении оцениваемых состояний, достижения своих целей.

**база знаний:** Структурированное хранилище информации, используемое интеллектуальным агентом для принятия решений, рассуждений и планирования в различных средах.

**большая языковая модель:** Разновидность искусственной нейронной сети, которая обучается на больших объемах данных для генерации и понимания естественного языка. Большая языковая модель использует технологии глубокого обучения для генерации текста, ответов на вопросы, переводов и многое другое.

**БЯМ-агент:** Интеллектуальный агент, воспринимающий информацию на естественном языке, знания которого хранятся в большой языковой модели и дополняются из контекста.

**дистанционные образовательные технологии:** Образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

**генерация с дополненной выборкой:** Алгоритм, при котором сначала происходит поиск информации в заданных источниках, найденная информация передается для дополнения контекста, на основе которого будет сгенерирован текст, являющийся ответом БЯМ.

**глубокое обучение нейронной сети:** Подход к созданию обширных иерархических представлений посредством обучения нейронных сетей с большим количеством скрытых слоев.

**индивидуальный учебный план:** Учебный план, обеспечивающий освоение образовательной программы на основе индивидуализации ее содержания с учетом особенностей и образовательных потребностей конкретного обучающегося.

**индивидуальная образовательная траектория:** Результат прохождения индивидуального учебного плана. Последовательность учебных задач и материалов, которая формируется исходя из анализа персональных характеристик обучающегося и целей обучения.

**инженерия знаний:** Дисциплина, рассматривающая приобретение знаний от специалистов в области знаний и из других источников знаний и включение их в базу знаний.

**интеллектуальная электронная образовательная система:** Электронная образовательная система, обладающая признаками искусственного интеллекта.

**интеллектуальный агент:** Агент, который выбирает действия, максимизирующие меру его эффективности, основываясь на воспринимаемой информации и знаниях, хранящихся в памяти

**искусственный интеллект ИИ:** Способность спроектированной системы приобретать, обрабатывать и применять знания и навыки.

**компетенция:** Динамическая комбинация знаний, умений и способность применять их для успешной профессиональной деятельности.

**контактная работа:** Форма образовательной деятельности, включающая занятия лекционного типа, занятия семинарского типа, групповые консультации, индивидуальную работу обучающихся с педагогическими работниками организации и иные формы взаимодействия. Контактная работа может проводиться с применением электронного обучения, дистанционных образовательных технологий.

**контекст:** ССовокупность ситуационных данных, предоставляемых агенту в рамках входного запроса, которые необходимы для точной интерпретации намерений пользователя, генерации релевантных ответов и поддержания связности диалога. Контекст включает историю сообщений, профиль пользователя, специфику задачи, внешние данные и другую информацию, не входящую в исходные обучающие данные модели. Набор данных или информации, который модель использует для генерации ответов или выполнения задач. Контекст помогает модели «понимать» текущую ситуацию, учитывать предыдущие взаимодействия или предоставленные данные, чтобы давать более релевантные и точные ответы.

**координация:** Для ИИ-агентов процесс организации взаимодействия агентов для достижения общей цели при заданных ограничениях

**машинное обучение:** Процесс оптимизации параметров модели с помощью вычислительных методов таким образом, чтобы поведение модели отражало данные и/или опыт

**многоагентная система:** Система, состоящая из множества взаимодействующих агентов. Эти агенты могут быть кооперативными (работают для достижения общей цели) или конкурентными (преследуют личные интересы). Их взаимодействие регулируется протоколами коммуникации и правилами принятия решений.

**нейронная сеть:** Сеть примитивных обрабатывающих элементов, соединенных взвешенными связями с регулируемыми весами, в которой каждый элемент выдает значение, применяя нелинейную функцию к своим входным значениям, и передает его другим элементам или представляет его в качестве выходного значения.

**нейронная сеть глубокого обучения:** Нейронная сеть обученная с помощью глубокого обучения

**модель компетенций:** В контексте ФГОС это систематизированный перечень требований к знаниям, умениям, навыкам и личностным качествам студента, которые он должен освоить в результате обучения. Она включает:

универсальные (общекультурные) компетенции, общепрофессиональные компетенции и профессиональные компетенции.

**образовательный модуль:** Совокупность учебных элементов, достаточных для организации и прохождения обучения по отдельной теме (занятию) дисциплины (курса).

**онтология:** Описание (подобно формальной спецификации программы) понятий и отношений, которые могут существовать для агента или сообщества агентов

**программный комплекс:** Программа, состоящая из двух или более компонентов и (или) комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого комплекса.

**система искусственного интеллекта:** Техническая система, использующая одну или несколько моделей ИИ, которая порождает такие конечные результаты, как контент, прогнозы, рекомендации или решения для заданного набора определенных человеком целей.

**технологии искусственного интеллекта:** Комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая самообучение и поиск решений без заранее заданного алгоритма) и получать результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека при решении задач компьютерного зрения, обработки естественного языка, распознавания и синтеза речи, поддержки принятия решений и других практически значимых задач обработки данных.

**уровень ЗУН:** Степень сформированности у обучающегося компетенций, определяемая через его способность решать задачи определённого класса.

**учебный план:** Документ, который определяет перечень, трудоемкость, последовательность и распределение по периодам обучения учебных предметов, курсов, дисциплин (модулей), практики, иных видов учебной деятельности и формы промежуточной аттестации обучающихся.

**электронная образовательная система:** Автоматизированная система, обеспечивающая процесс электронного обучения.

**электронное обучение:** Организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников. На практике информационные технологии, обеспечивающие обработку используемой при реализации образовательных программ информации, реализуются путем создания автоматизированных систем

**языковая модель:** Вероятностная модель, предсказывающая следующее слово в последовательности на основе предыдущих слов

## Список литературы

### Книги и монографии

1. Мирошниченко, М.А. Цифровая трансформация: российские приоритеты формирования цифровой экономики : монография / М.А. Мирошниченко. – Краснодар : Издательство Кубанского государственного университета, 2021. – 224 с. – 500 экз. – ISBN 978-5-8209-2005-9.

2. Рассел, С. Искусственный интеллект. Современный подход : в 3 томах. Том 3. 4-е издание / С. Рассел, П. Норвиг. – Москва : Диалектика, 2022. – 638 с. – ISBN 978-5-907365-27-8.

3. Ларичев О.И. Теория и методы принятия решений, а также хроника событий в волшебных странах / О.И. Ларичев. – 3-е издание. – Москва : Издательство Логос, 2006. – 392 с. – ISBN 5-88439-046-7.

### Нормативные правовые акты

4. Об утверждении профессионального стандарта «Специалист по тестированию в области информационных технологий [приказ Министерства труда и социальной защиты Российской Федерации от 2 августа 2021 г. № 531н]. – Справочно-правовая система «Консультант Плюс». – Текст : электронный. – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_394702/](https://www.consultant.ru/document/cons_doc_LAW_394702/) (дата обращения: 21.11.2025).

5. Об утверждении профессионального стандарта «Программист» [приказ Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 года № 424н]. – Справочно-правовая система «Консультант Плюс». – Текст : электронный. – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_425094/](https://www.consultant.ru/document/cons_doc_LAW_425094/) (дата обращения: 21.11.2025).

6. Российская Федерация. Законы. Об образовании в Российской Федерации : федеральный закон [принят Государственной Думой 29 декабря 2012 г.]/ Справочно-правовая система «Консультант Плюс».

– Текст : электронный. – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](https://www.consultant.ru/document/cons_doc_LAW_140174/) (дата обращения: 21.11.2025).

7. Методические рекомендациями по разработке и внедрению инновационных технологий в реализацию дополнительных профессиональных программ, в том числе по созданию сети симуляционных центров (виртуальных лабораторий) [Письмо Минпросвещения России от 01.02.2021 № 08-166]. – Справочно-правовая система «Консультант Плюс».

– Текст : электронный. – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_427000/96c60c11ee5b73882df84a7de3c4fb18f1a01961/](https://www.consultant.ru/document/cons_doc_LAW_427000/96c60c11ee5b73882df84a7de3c4fb18f1a01961/) (дата обращения: 21.11.2025).

### Стандарты

8. ГОСТ Р 59899-2021 Образовательные продукты с алгоритмами искусственного интеллекта для адаптивного обучения в общем образовании. Технические требования: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию от 26 ноября 2021 г. № 1621-ст: введен впервые : дата введения 2022-03-01 / разработан Федеральным государственным автономным образовательным учреждением высшего образования «Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ). – Москва : Российский институт стандартизации, 2021.

9. ГОСТ Р 70462.1-2022/ISO/IEC TR 24029-1-2021 Информационные технологии (ИТ). Интеллект искусственный. Оценка робастности нейронных сетей. Часть 1 = Artificial Intelligence (AI) – Assessment of the robustness of neural networks - Part 1: Overview: Обзор: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию от 2 ноября 2022 г. № 1226-ст: введен впервые : дата введения 2022-11-02 / разработан Федеральным государственным автономным образовательным учреждением

высшего образования «Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ). – Москва : Российский институт стандартизации, 2022.

### Диссертации

10. Аль-Дулаими, О.Х.З. Интеллектуализация процессов принятия решений в системе управления качеством подготовки обучаемых в организационных корпоративных системах на основе нейросетевых моделей представления знаний : специальность 2.3.4. Управление в организационных системах : диссертация на соискание ученой степени кандидата технических наук / Аль-Дулаими Омар Хатем Заидан ; Воронежский государственный технический университет. – Воронеж, 2024. – 132 с. – Библиогр.: с. 119-130.  
 – Текст : электронный. – URL: <https://cchgeu.ru/science/dissertatsionnye-sovety/dissertatsionnyy-sovet-d-212-037-13/soiskateli/al-dulaimi.php> (дата обращения: 30.11.2025).

11. Гнедых, Д.С. Эффективность усвоения учебной информации студентами в условиях электронного обучения : специальность 19.00.07 Педагогическая психология : диссертация на соискание ученой степени кандидата психологических наук / Гнедых Дарья Сергеевна ; Санкт-Петербургский государственный университет. – Санкт-Петербург, 2015. – 237 с. – Библиогр.: с. 173-203. – Текст : электронный. – URL: [https://disser.spbu.ru/files/disser2/disser/Gnedykh\\_Daria\\_disser.pdf](https://disser.spbu.ru/files/disser2/disser/Gnedykh_Daria_disser.pdf) (дата обращения: 30.11.2025).

12. Косоногова, М.А. Метод и средства управления образовательной траекторией в системах электронного обучения : специальность 05.13.10 Управление в социальных и экономических системах : диссертация на соискание ученой степени кандидата технических наук / Косоногова Марина Александровна ; Юго-Западный государственный университет. – Курск, 2016. – 160 с. – Библиогр.: с. 163-180. – Текст :

электронный. – URL: <http://www.swsu.ru/ds/diss-swsu/> (дата обращения: 30.09.2025).

13. Кречетов, И.А. Модели, алгоритмы и инструментальные средства адаптивного обучения : специальность 05.13.10 Управление в социальных и экономических системах : диссертация на соискание ученой степени кандидата технических наук / Кречетов Иван Анатольевич ; Томский государственный университет систем управления и радиоэлектроники. – Томск, 2021. – 150 с. – Библиог.: с. 99-106. – Текст : электронный. – URL: <https://postgraduate.tusur.ru/urls/h80g08x8> (дата обращения: 30.09.2025).

14. Нгуен, В.Н. Методы и алгоритмы структуризации образовательного контента и управления процессом электронного обучения персонала промышленных предприятий на основе графовых моделей : специальность 05.13.01 Системный анализ, управление и обработка информации : диссертация на соискание ученой степени кандидата технических наук / Нгуен Ван Нгон ; Южно-Российский государственный политехнический университет (НПИ) им. М. И. Платова. – Новочеркасск, 2017. – 257 с. – Библиог.: с. 168-185. – Текст : электронный. – URL: [http://www.npi-tu.ru/assets/diss/d212.304.02/files/ngon/diss\\_ngon.pdf](http://www.npi-tu.ru/assets/diss/d212.304.02/files/ngon/diss_ngon.pdf) (дата обращения: 30.11.2025).

15. Нгуен, Ф.Х. Модели и алгоритмы обработки информации для поддержки принятия решений при управлении программами дополнительного профессионального образования персонала промышленных предприятий : специальность 05.13.01 Системный анализ, управление и обработка информации : диссертация на соискание ученой степени кандидата технических наук / Нгуен Фук Хай ; Южно-Российский государственный политехнический университет (НПИ) им. М.И. Платова. – Новочеркасск, 2021. – 156 с. – Библиог.: с. 130-147. – Текст : электронный. – URL: [https://www.npi-tu.ru/assets/diss/d212.304.02/files/hau/diss\\_nguyen\\_phuc\\_hau.pdf](https://www.npi-tu.ru/assets/diss/d212.304.02/files/hau/diss_nguyen_phuc_hau.pdf) (дата обращения: 30.11.2025).

16. Николаев, И.Е. Методы и алгоритмы интеллектуальной поддержки формирования требований вакансии на основе нейросетевых моделей языка и актуальных требований рынка труда : специальность 2.3.4. Управление в организационных системах : диссертация на соискание ученой степени кандидата технических наук / Николаев Иван Евгеньевич ; Уфимский университет науки и технологий. – Уфа, 2025. – 162 с. – Библиогр.: с. 142-155.  
– Текст : электронный. – URL: <https://uust.ru/media/dc/24247902/234/nikolaev-ivan-evgenevich-2024-10-08/dissertation.pdf>  
(дата обращения: 30.11.2025).

#### Авторефераты диссертаций

17. Амосов, О.С. Нейросетевые и нечеткие методы оценивания стохастических систем : специальность 05.13.18 Математическое моделирование, численные методы и комплексы программ : автореферат диссертации на соискание ученой степени доктора технических наук / Амосов Олег Семенович ; ГОУВПО Комсомольский-на-Амуре государственный технический университет. – г. Комсомольск-на-Амуре, 2004. – 37 с – Библиогр. с.: 32-37. – Место защиты: Институт автоматики и проблем управления ДВО РАН.

18. Ботов, Д.С. Методы и алгоритмы интеллектуальной поддержки формирования образовательных программ по требованиям рынка труда на основе нейросетевых моделей языка : специальность 05.13.10 Управление в социальных и экономических системах : автореферат диссертации на соискание ученой степени кандидата технических наук / Ботов Дмитрий Сергеевич ; Челябинский государственный университет. – Челябинск, 2019. – 17 с – Библиогр.: с. 15-17. – Место защиты: ФГБОУ ВО Уфимский государственный авиационный технический университет.

19. Власенко, А.А. Разработка адаптивной системы дистанционного обучения в сфере информационных технологий : специальность 05.13.17 Теоретические основы информатики : автореферат диссертации на соискание

ученой степени кандидата технических наук /  
 Власенко Алексей Александрович ; Воронежский государственный университет. – Воронеж, 2014. – 16 с – Библиогр.: с. 14-15. – Место защиты: ФГБОУ ВО Воронежский государственный университет.

20. Макаров, А.В. Методы, алгоритмы и система управления процессом обучения на основе семантико-логического анализа учебной информации : специальность 05.13.10 Управление в социальных и экономических системах : автореферат диссертации на соискание ученой степени кандидата технических наук / Макаров Андрей Владимирович ; Рязанский государственный радиотехнический университет имени В. Ф. Уткина. – Рязань, 2013. – 22 с – Библиогр.: с. 19-21. – Место защиты: Рязанский государственный радиотехнический университет имени В.Ф. Уткина.

21. Махныткина, О.В. Моделирование и оптимизация индивидуальной траектории обучения студента : специальность 05.13.10 Управление в социальных и экономических системах : автореферат диссертации на соискание ученой степени кандидата технических наук / Махныткина Олеся Владимировна ; Сибирский государственный университет телекоммуникаций и информатики. – Новосибирск, 2013. – 23 с – Библиогр.: с. 20-23. – Место защиты: Сибирский государственный университет телекоммуникаций и информатики.

22. Медведев, Р.Е. Алгоритмы автоматизированного формирования баз знаний для систем дистанционного обучения : специальность 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей : автореферат диссертации на соискание ученой степени кандидата технических наук / Медведев Роман Евгеньевич ; Рязанский государственный радиотехнический университет имени В.Ф. Уткина. – Рязань, 2013. – 16 с – Библиогр.: с. 14-16. – Место защиты: Рязанский государственный радиотехнический университет имени В.Ф. Уткина.

23. Перченок, О.В. Модели и алгоритмы верификации решений задач в системах электронного обучения : специальность 05.13.06 Автоматизация и управление технологическими процессами и производствами : автореферат диссертации на соискание ученой степени кандидата технических наук / Перченок Олег Владимирович ; Национальный исследовательский университет ИТМО. – Санкт-Петербург, 2013. – 18 с – Библиогр.: с. 17-18. – Место защиты: Национальный исследовательский университет ИТМО.

24. Сеньковская, А.А. Математическое моделирование и методы оптимизации учебного процесса вуза (на примере задач формирования рабочих учебных планов) : специальность 05.13.18 Математическое моделирование, численные методы и комплексы программ: автореферат диссертации на соискание ученой степени кандидата технических наук / Сеньковская Анастасия Александровна ; Омский государственный университет им. Ф.М. Достоевского. – Омск, 2021. – 20 с. – Библиогр.: с. 18-20. – Место защиты: Омский государственный технический университет.

25. Ульянов, Д.А. Марковская модель адаптивного тестирования и ее программная реализация в условиях дистанционного обучения : специальность 05.13.18 Математическое моделирование, численные методы и комплексы программ : автореферат диссертации на соискание ученой степени кандидата технических наук / Ульянов Дмитрий Александрович ; Иркутский государственный технический университет. – Иркутск, 2005. – 16 с. – Библиогр.: с. 15-16. – Место защиты: Байкальский государственный университет экономики и права.

26. Шана, М.А. Автоматизированная информационная система адаптивного обучения на основе компетентностного подхода : специальность 05.25.05 Информационные системы и процессы : автореферат диссертации на соискание ученой степени кандидата технических наук / Шана Маха Акрам ; Кабардино-Балкарский государственный университет им. Х. М. Бербекова. – Нальчик, 2014. – 22 с. – Библиогр.: с. 20-22. – Место защиты: Южный федеральный университет.

### Электронные ресурсы

27. Адаптивное обучение в вузе: что это такое, методы, этапы, платформы / Zaochnik : сайт. – Текст : электронный. – URL: <https://zaochnik.ru/blog/adaptivnoe-obuchenie-v-vuze-chto-eto-takoe-metody-etapy-platformy/> (дата обращения: 22.09.2025).
28. Соболевская, О. Картотека: адаптивное обучение / О. Соболевская // IQMedia : сайт. – ISSN отсутствует. – Текст : электронный. – DOI отсутствует. – URL: <https://iq-media.ru/archive/428632186> (дата обращения: 22.09.2025).
29. Хуторской, А.В. Индивидуализация и профильность обучения в старшей школе / А.В. Хуторской // khutorskoy : сайт. – ISSN отсутствует. – Текст : электронный. – DOI отсутствует. – URL: <https://www.khutorskoy.ru/books/bibliography-e.htm> (дата обращения: 22.09.2025).
30. Bechard, P. Reducing hallucination in structured outputs via Retrieval-Augmented Generation / P. Bechard // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2404.08189>. – URL: <https://arxiv.org/abs/2404.08189> (дата обращения: 22.09.2025).
31. Bhowmick, A. On the Cut Locus of Submanifolds of a Finsler Manifold / A. Bhowmick // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2307.11045>. – URL: <https://arxiv.org/abs/2307.11045> (дата обращения: 22.09.2025).
32. Brookhart, S. A Century of Grading Research: Meaning and Value in the Most Common Educational Measure / S. Brookhart // Sage Journals. – ISSN отсутствует. – Текст : электронный. – DOI <https://doi.org/10.3102/0034654316672069>. – URL: <https://journals.sagepub.com/doi/10.3102/0034654316672069> (дата обращения: 22.09.2025).

33. Cheung, K. Interpretation of excess in H -> Zy using a light axion-like particle / K. Cheung // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный.  
– DOI <https://doi.org/10.48550/arXiv.2402.05678>. – URL: <https://arxiv.org/abs/2402.05678> (дата обращения: 22.09.2025).
34. Cyrus, R. Deeper insights into retrieval augmented generation: The role of sufficient context / R. Cyrus // Google Research : сайт. – ISSN отсутствует.  
– Текст : электронный. – DOI отсутствует. – URL: <https://research.google/blog/deeper-insights-into-retrieval-augmented-generation-the-role-of-sufficient-context/> (дата обращения: 22.09.2025).
35. Edge, D. From Local to Global: A Graph RAG Approach to Query-Focused Summarization / D. Edge // arxiv.org : сайт. – ISSN 2331-8422.  
– Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2404.16130>. – URL: <https://arxiv.org/abs/2404.16130> (дата обращения: 22.09.2025).
36. Hong, S. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework / S. Hong // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2308.00352>. – URL: <https://arxiv.org/abs/2308.00352> (дата обращения: 22.09.2025).
37. Lewis, P. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2005.11401>. – URL: <https://arxiv.org/abs/2005.11401> (дата обращения: 22.09.2025).
38. Nguen, T. Jailbreak Detection in Clinical Training LLMs Using Feature-Based Predictive Models / T. Nguen // arxiv.org : сайт. – ISSN 2331-8422.  
– Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2505.00010>. – URL: <https://arxiv.org/abs/2505.00010> (дата обращения: 22.09.2025).
39. Orang, O. Causal Graph Fuzzy LLMs: A First Introduction and Applications in Time Series Forecasting / O. Orang // arxiv.org : сайт.  
– ISSN 2331-8422. – Текст : электронный.  
– DOI <https://doi.org/10.48550/arXiv.2507.17016>. – URL: <https://arxiv.org/abs/2507.17016> (дата обращения: 22.09.2025).

40. Quinn, L. Long Context RAG Performance of Large Language Models / L. Quinn // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2411.03538v1>. – URL: <https://arxiv.org/html/2411.03538v1> (дата обращения: 22.09.2025).

41. Wu, Q. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation / Q. Wu // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2308.08155>. – URL: <https://arxiv.org/abs/2308.08155> (дата обращения: 22.09.2025).

42. Zheng, W. LLM-as-a-Fuzzy-Judge: Fine-Tuning Large Language Models as a Clinical Evaluation Judge with Fuzzy Logic / W. Zheng, T. Laurah, K. Jess., [et al.] // arxiv.org : сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2506.11221>. – URL: <https://arxiv.org/abs/2506.11221> (дата обращения: 22.09.2025).

43. Enhance your adaptive teaching with a new Moodle plugin / Moodle : официальный сайт. – Текст : электронный. – URL: <https://moodle.com/news/enhance-your-adaptive-teaching-with-a-new-moodle-plugin/> (дата обращения 04.11.2025).

44. AI providers / Moodle : официальный сайт. – Текст : электронный. – URL: [https://docs.moodle.org/501/en/AI\\_providers](https://docs.moodle.org/501/en/AI_providers) (дата обращения 04.11.2025).

45. Explore the latest AI solutions for Moodle / Moodle : официальный сайт. – Текст : электронный. – URL: <https://moodle.com/us/news/ai-solutions-for-moodle> (дата обращения 04.11.2025).

## Статьи

46. Алтухов, А.И. Метод оценивания качества применения учебно-тренажерного комплекса при проведении практических занятий / А. И. Алтухов, С.А. Багрецов, М.А. Чебурков // Известия СПбГЭТУ ЛЭТИ. – 2018. – № 1. – С. 31–37. – ISSN 2071-8985.

47. Андриевская, Н.В. Особенности применения нейро-нечетких моделей для задач синтеза систем автоматического управления / Н.В. Андриевская, А.С. Резников, А.А. Черанев // Фундаментальные исследования. – 2014. – № 11–7. – С. 1445–1449 – ISSN 1812-7339.
48. Багрецов, С.А. Использование дискриминантного анализа для агрегирования оценок знаний обучающихся в номинальных шкалах / С.А. Багрецов, Э.В. Мищенко, Л.В. Розанова // Эргодизайн. – 2023. – №1 (19). – С. 43–52. – ISSN 2658-4026.
49. Багрецов, С.А. Метод структурно-логического анализа ошибок обучаемых в контролирующих процедурах автоматизированных обучающих системах / С.А. Багрецов, В.М. Калуга // Известия Тульского государственного университета. Технические науки. – 2022. – № 6. – С. 97–100.– ISSN 2071-6168.
50. Багрецов, С.А. Методика построения плана диагностического исследования профессионального соответствия кандидатов в системах профессионального отбора / С.А. Багрецов, Э.В. Мищенко, Л.В. Розанова // Эргодизайн. – 2022. – № 4 (18). – С. 243–251. – ISSN 2658-4026.
51. Багрецов, С.А. Модель трансформационного процесса индивидуально-группового обучения в вузе / С.А. Багрецов, Т.Э. Черная // Труды Военно-космической академии им. А.Ф. Можайского. – 2019. – № 667. – С. 400–409. – ISSN 2218-5429.
52. Багрецов, С.А. Система отбора кандидатов и подготовки специалистов в системе военного образования с учетом распределения имеющихся ресурсов / С.А. Багрецов, Э.В. Мищенко // Труды Военно-космической академии им. А.Ф. Можайского. – 2013. – № 639. – С. 163–169. – ISSN 2218-5429.
53. Благов, Е.Ю. Онтологический подход в практике образовательной деятельности: формирование траекторий индивидуального профессионального развития студентов / Е.Ю. Благов, И.А. Лещева, С.А. Щербан // Открытое образование. – 2018. – № 5. Том 2. – С. 26–39. – ISSN 1818-4243.

54. Борисенко, Н.А. «Барометр влияния», или Какие факторы оказывают наибольшее воздействие на обучение. Рецензия на книгу: Джон Хэтти «Видимое обучение» / Н.А. Борисенко // Вопросы образования. – 2018. – № 1. – С. 257–264. – ISSN 1814-9545.

55. Васильев, А.Ф. Построение нечеткой оценки компетентности учителя математики в среде Matlab Fuzzy Logic Toolbox / А.Ф. Васильев, А.С. Вегера, Е.Н. Мысловец // Современное образование: преемственность и непрерывность образовательной системы «школа – вуз» : материалы VII Международной научно-методической конференции – Гомель : Издательство ГГУ им. Ф. Скорины, 2009. – С. 174–176. – ISBN отсутствует.

56. Гаспариан, М.С. Инжиниринг образовательных программ на основе применения интеллектуальных технологий / М.С. Гаспариан, С.А. Лебедев, Ю.Ф. Тельнов // Открытое образование. – 2017. – № 1. Том 2. – С. 14–19. – ISSN 1818-4243.

57. Дармахеева, Т.А. Онтологическая модель представления знаний о системе продукции / Т.А. Дармахеева, Н.Б. Хаптахаева // Знания – Онтологии – Теории (ЗОНТ-2017) : материалы всероссийской конференции с международным участием – Новосибирск : Дигит Про, 2017. – С. 119–128. – ISBN отсутствует.

58. Евдокимова, И.С. Алгоритмы трансляции естественно-языковых запросов к СУБД MYSQL, Freebase и dbpedia через промежуточный язык / И. С. Евдокимова, О. И. Буденная // Теоретические и прикладные вопросы современных информационных технологий. – Улан-Удэ : Издательство ВСГУТУ, 2015. – С. 95–104. – ISBN отсутствует.

59. Евдокимова, И.С. Математическая модель трансляции естественно-языкового запроса с использованием абстрактного языка запросов / И.С. Евдокимова, А.Н. Тимофеев // В мире научных открытий. – 2014. – № 12–1 (60). – С. 374–393. – ISSN 2072-0831.

60. Евдокимова, И.С. Сравнительный анализ языков запросов к структурированным источникам данных с целью разработки промежуточного языка представления ЕЯ-запросов / И.С. Евдокимова, А.Н. Тимофеев // Вестник ВСГУТУ. – 2013. – № 6 (45). – С. 61–68. – ISSN 2413-1997.
61. Ермаков, С.С. Современные технологии электронного обучения анализ влияния методов геймификации на вовлеченность учащихся в образовательный процесс / С.С. Ермаков // Современная зарубежная психология. – 2020. – № 3. Том 9 – С. 47–58. – ISSN 2304-4977.
62. Зарипова, Е.И. Индивидуальная образовательная программа студента как средство индивидуализации магистерской подготовки / Е.И. Зарипова // Вестник Омского государственного педагогического университета. Гуманитарные исследования. – 2014. – № 1 (2). – С. 124–128. – ISSN 2309-9380.
63. Кузнецов, А.В. Краткий обзор многоагентных моделей / А.В. Кузнецов // Управление большими системами: сборник трудов. – 2018. – № 71. – С. 6–44. – ISSN отсутствует.
64. Лиференко, В.Д. Методики агрегирования оценок знаний обучающихся для случаев представления внешнего критерия в порядковых шкалах / В.Д. Лиференко, С.А. Багрецов, Д.В. Чистяков // Наукоемкие технологии в космических исследованиях Земли. – 2021. – № 2. Том 13. – С. 4-14. – ISSN 2412-1363.
65. Малинин, Д.А. Прогнозирование успеваемости студентов на основе нечеткой логики / Д.А. Малинин, Л.А. Станкевич // Top Technologies. – 2017. – № 5. – С. 1–10. – ISSN 1812-7320.
66. Бровка, Н.В. Марковская математическая модель динамического адаптивного тестирования активного агента / Н.В. Бровка, П.П. Дьячук, М.В. Носков, И.П. Перегудова // Информатика и образование. – 2018. – № 10 (299). – С. 29–35. – ISSN 0234-0453.
67. Багрецов, С.А. Методика априорной оценки эффективности подготовки операторов в системах управления / С.А. Багрецов, О.С. Лаута,

Е.Г. Баленко [и др.] // Известия Тульского государственного университета. Технические науки. – 2023. – № 8. – С. 124–131. – ISSN 2071-6168.

68. Минасов, Ш.М. Проект «Гефест» как вариант практической реализации технологий электронного обучения в вузе в условиях интеграции традиционного и дистанционного обучения / Ш.М. Минасов, С.В. Тархов // Образовательные технологии и общество. – 2005. – № 1. Том 8. – С. 134–147. – ISSN 1436-4522.

69. Минасова, Н.С. Модели и методы адаптивного управления обучением с использованием агрегативных учебных модулей / Н.С. Минасова, С.В. Тархов, Л.М. Тархова // Вестник Уфимского государственного авиационного технического университета. – 2017. – № 4 (78). Том 21. – С. 136-142. – ISSN 2225-2789.

70. Мищенко, Э.В. Методика выбора оптимальных вариантов подготовки военных специалистов (на этапе ввода в строй) с применением векторной оптимизации / Э.В. Мищенко // Инновационная деятельность в вооруженных Силах Российской Федерации : труды всеармейской научно-практической конференции. – Санкт-Петербург : Издательство Военной академии связи имени маршала Советского Союза С. М. Буденного Министерства обороны РФ, 2015. – С. 309-318. – ISBN отсутствует.

71. Бородин, В.В. Многоагентная динамическая модель многокритериального информационного взаимодействия структурных элементов самоорганизующейся сети передачи данных наземно-воздушной системы мониторинга / В.В. Бородин, Д.А. Клецков, А.В. Тимошенко, В.А. Щевцов // Известия Российской академии наук. Теория и системы управления. – 2023. – № 1. – С. 123–136. – ISSN 0002-3388.

72. Исаков, А.О. Особенности организации игрового взаимодействия асимметричных агентов с использованием графовых нейронных сетей / А.О. Исаков, Д.Е. Перегородиев, И.В. Томилов [и др.] // Научно-технический вестник информационных технологий, механики и оптики. – 2024. – № 6. Том 24. – С. 1044–1048. – ISSN 2226-1494.

73. Павлов, Д.А. Нечеткий фильтр Калмана в структуре алгоритма решения обратных задач для экономических объектов / Д.А. Павлов // Программные продукты и системы. – 2013. – № 4. – С. 36. – ISSN 0236-235X.
74. Поздеева, С.И. Рецензия на книгу Джона А.С. Хэтти. Видимое обучение: синтез результатов более 50 000 исследований с охватом более 86 миллионов школьников / С.И. Поздеева // Вестник ТГПУ. – 2020. – № 1 (207). – ISSN 1609-624X. – Текст : электронный. – DOI отсутствует. – URL: <https://cyberleninka.ru/article/n/retsenziya-na-knigu-vidimoe-obuchenie-sintez-rezultatov-bolee-50-000-issledovaniy-s-ohvatom-bolee-86-millionov-shkolnikov-dzhon-a-s-hetti-pod> (дата обращения: 03.05.2025).
75. Полетайкин, А.Н. Технология разработки и верификации профессиональных стандартов, их применения в системах управления обучением на основе онтологий / А.Н. Полетайкин, С.Г. Синица, Е.Ю. Кунц // Экономика и управление: теория и практика. – 2020. – № 2. Том 6. – С. 37–46. – ISSN 2412-8376.
76. Поминов, Д.А. Самообучающийся адаптивный тренажер для подготовки абитуриентов к экзаменам по математике / Д.А. Поминов // Нейрокомпьютеры: разработка, применение. – 2021. – № 2. Том 23. – С. 35-42. – ISSN 1999-8554.
77. Козлова, Л.Я. Развитие современной образовательной среды высших военно-инженерных учебных заведений с использованием искусственного интеллекта / Л.Я. Козлова, О.В. Крылова, Е.К. Яхваров, Е.О. Кудро // Высшее образование сегодня. – 2024. – № 1. – С. 50–59. – ISSN 1726-667X.
78. Багрецов, С.А. Разработка методов профессиографических исследований при проведении кадровой политики предприятий / С.А. Багрецов, В.М. Львов, Ю.М. Органов [и др.] // Человеческий фактор: проблемы психологии и эргономики. – 2006. – № 4 (37). – С. 52–56. – ISSN 2415-7023.

79. Тельнов, Ю.Ф. Разработка прототипа многоагентной системы сетевого взаимодействия учебных заведений / Ю.Ф. Тельнов, А.В. Данилов, Р.И. Дивеев [и др.] // Открытое образование. – 2018. – № 22(6). – С. 14–26. – ISSN 1818-4243.
80. Тимофеев, А.Н. Семантическое моделирование: обзор процессов, инструментов, методов и знаний предметной области разработки программного обеспечения (Часть 1) / А.Н. Тимофеев, И. С. Евдокимова, Н.Б. Хаптакаева, А.А. Сенотрусова // Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки». – 2022. – № 12. – С. 85-91. – ISSN 2223-2966.
81. Соколова, Ю.А. Генезис понятия «адаптация» / Ю.А. Соколова // Интеллект. Инновации. Инвестиции. – 2014. – № 3. – ISSN 2713-1424. – Текст : электронный. – DOI отсутствует. – URL: <https://cyberleninka.ru/article/n/genezis-ponyatiya-adaptatsiya> (дата обращения: 02.05.2025).
82. Тарасов, В.Б. Нетрадиционные и гибридные логики в моделировании интеллектуальных агентов. I. Искусственные деятели, интенциональные характеристики и пути моделирования / В.Б. Тарасов // Известия РАН. Теория и системы управления. – 2000. – № 5. – С. 5–17. – ISSN 0002-3388.
83. Тимофеев, А.Н. Модель взаимодействия участников учебного процесса для формирования индивидуальной образовательной траектории с учётом деструктивных факторов / А.Н. Тимофеев, А.А. Главинская // Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки». - 2025. - № 9. - С. 105-114. – ISSN 2223-2966.
84. Тимофеев, А.Н. Разработка модели компетенций в сфере информационных технологий на основе интеграции онтологий и баз знаний / А.Н. Тимофеев // Современная наука: актуальные проблемы теории и

практики. Серия «Естественные и технические науки». – 2023. – № 6-2. – С. 153-159. – ISSN 2223-2966.

85. Тимофеев, А.Н. Семантическое моделирование: обзор процессов, инструментов, методов и знаний предметной области разработки программного обеспечения (Часть 2) / А.Н. Тимофеев, И.С. Евдокимова, Н.Б. Хаптахаева // Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки». – 2023. – № 4-2. – С. 122-129. – ISSN 2223-2966.

86. Тимофеев, А.Н. Разработка подхода к повышению качества генерации программного кода большими языковыми моделями / А.Н. Тимофеев, С.С. Михайлова // Нейрокомпьютеры: разработка, применение. – 2024. – № 4. Том 26. – С. 18-26. – ISSN 1999-8554.

87. Толмачев, А.В. Вероятности распределения успеваемости студентов онлайн курсов как инструмент оценки качества контрольных материалов / А.В. Толмачев, Е.В. Синицын, Г.В. Астратова // Отходы и ресурсы. – 2020. – № 3. Том 7. – С. 9. – ISSN 2500-0659.

88. Усикова, Ю.А. Разработка нечеткой модели оценивания компетенций студентов в соответствии с профессиональными стандартами / Ю.А. Усикова // Научный Лидер. – 2021. – № 40 (42). – С. 17–25. – ISSN 2713-3168.

89. Бикмухаметов, И.К. Факторы обеспечения качества электронного обучения в вузе / И.Х. Бикмухаметов, Е.А. Колганов, Р.Б. Шайхисламов, Н.Р. Сагманова // Управление экономическими системами: электронный научный журнал. – 2012. – № 11 (47). – С. 62. – ISSN 1999-4516.

90. Шамсутдинова, Т.М. Формирование индивидуальной образовательной траектории в адаптивных системах управления обучением / Т.М. Шамсутдинова // Открытое образование. – 2021. – № 25 (6). – С. 36–44. – ISSN 1818-4243.

91. Asim, M. N. A survey of ontology learning techniques and applications / M. N. Asim, M. Wasim, M. U. G. Khan [et al.] // Database: The Journal of Biological Databases and Curation. – 2018. – ISSN 1758-0463. – Текст : электронный. – DOI <https://doi.org/10.1093/database/bay101>. – URL: <https://www.semanticscholar.org/paper/A-survey-of-ontology-learning-techniques-and-Asim-Wasim/0047a6037303cb8507506c260a79bec99d280f21> (дата обращения: 19.06.2025).
92. Biesta, G. Good Education in an Age of Measurement: Ethics, Politics, Democracy / G. Biesta // arxiv.org : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2403.19506.2024>. – URL: <https://arXiv:2403.19506.2024> (дата обращения: 19.11.2025).
93. Chen, C. LLMs as Academic Reading Companions: Extending HCI Through Synthetic Personae / C. Chen, A. Leitch // arxiv.org : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2403.19506.2024>. – URL: <https://arXiv:2403.19506.2024> (дата обращения: 19.11.2025).
94. Cheng, L. Teachers' grading decisions and practices across cultures: Exploring the value, consistency, and construction of grades across Canadian and Chinese secondary schools / L. Cheng // ScienceDirect : официальный сайт. – ISSN 0191-491X. – Текст : электронный. – DOI <https://doi.org/10.1016/j.stueduc.2020.100928>. – URL: <https://www.sciencedirect.com/science/article/pii/S0191491X20301760> (дата обращения: 03.12.2025).
95. Chen, J. Learning Programming Agent-based Modeling with LLM Companions: Experiences of Novices and Experts Using ChatGPT & NetLogo Chat / J. Chen [et al.] // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2401.17163.2024>. – URL: <https://arXiv:2401.17163.2024> (дата обращения: 19.11.2025).

96. Cheng, X. Adaptive learning interventions for students with dyslexia: A multilevel analysis / X. Cheng, J. Liu, H. Wen // Journal of Educational Psychology. – 2020. – №112 (4). – P.789–805. – ISSN 0022-0663.
97. Database management systems / Dbpedia : сайт. – Текст : электронный. – URL: [https://dbpedia.org/describe?url=http%3A%2F%2Fdbpedia.org%2Fresource%2FCategory%3ADatabase\\_management\\_systems&sid=15873](https://dbpedia.org/describe?url=http%3A%2F%2Fdbpedia.org%2Fresource%2FCategory%3ADatabase_management_systems&sid=15873) (дата обращения: 24.06.2025).
98. Denny, P. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language / P. Denny, V. Kumar, N. Giacaman // Proceedings of the 54th ACM Technical Symposium on Computer Science Education – 2023. – P. 1136–1142. – ISBN отсутствует. – Текст : электронный. – DOI <https://doi.org/10.1145/3545945.3569823>. – URL: [https://www.researchgate.net/publication/368989512\\_Conversing\\_with\\_Copilot\\_Exploring\\_Prompt\\_Engineering\\_for\\_Solving\\_CS1\\_Problems\\_Using\\_Natural\\_Language](https://www.researchgate.net/publication/368989512_Conversing_with_Copilot_Exploring_Prompt_Engineering_for_Solving_CS1_Problems_Using_Natural_Language) (дата обращения: 19.11.2025).
99. Denny, P. Prompt Problems: A New Programming Exercise for the Generative AI Era / P. Denny [et al.] // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2311.05943.2023> (дата обращения: 19.11.2025).
100. Dublin Core specifications / Dublin Core : официальный сайт. – Текст : электронный. – URL: <https://www.dublincore.org/specifications/dublin-core/> (дата обращения: 19.06.2025).
101. Dunder, N. «Kattis» vs. ChatGPT: Assessment and Evaluation of Programming Tasks in the Age of Artificial Intelligence / N. Dunder // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2312.01109.2023> (дата обращения: 19.11.2025).
102. Denny, P. Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators / P. Denny, J. Leinonen, J. Prather [et al.]

// arXiv. : официальный сайт. – ISSN 2331-8422. – Текст : электронный.  
– DOI <https://doi.org/10.48550/arXiv.2307.16364>. – URL:  
<https://arxiv.org/abs/2307.16364> (дата обращения: 11.11.2025).

103. Gruber, Th. What is an Ontology / Th. Gruber // Stanford-edu :  
официальный сайт. – ISSN отсутствует. – Текст : электронный.  
– DOI отсутствует. – URL: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> (дата обращения: 05.04.2025).

104. Hattie, J. Visible Learning and the Science of How We Learn / J. Hattie,  
G. C.R. Yates // inspirasifoundation : сайт. – ISSN отсутствует. – Текст :  
электронный. – DOI 10.4324/9781315885025. – URL:  
[https://inspirasifoundation.org/wp-content/uploads/2020/05/John-Hattie-Visible-Learning\\_-A-synthesis-of-over-800-meta-analyses-relating-to-achievement-2015.pdf](https://inspirasifoundation.org/wp-content/uploads/2020/05/John-Hattie-Visible-Learning_-A-synthesis-of-over-800-meta-analyses-relating-to-achievement-2015.pdf) (дата обращения: 30.11.2025).

105. Hattie, J. Visible Learning: A Synthesis of Over 800 Meta-Analyses  
Relating to Achievement / J. Hattie // inspirasifoundation : сайт.  
– ISBN 0-203-88733-6 – Текст : электронный. – DOI отсутствует. – URL:  
[https://inspirasifoundation.org/wp-content/uploads/2020/05/John-Hattie-Visible-Learning\\_-A-synthesis-of-over-800-meta-analyses-relating-to-achievement-2008.pdf](https://inspirasifoundation.org/wp-content/uploads/2020/05/John-Hattie-Visible-Learning_-A-synthesis-of-over-800-meta-analyses-relating-to-achievement-2008.pdf) (дата обращения: 30.11.2025).

106. Hoffman, L. Persons as contexts: Evaluating between-person and  
within-person effects in longitudinal analysis / L. Hoffman, R. S. Stawski //  
Research in Human Development. – 2009. – 6 (2–3). – P.97–120. – ISSN 15427609.

107. Kerslake, C. Integrating Natural Language Prompting Tasks in  
Introductory Programming Courses / C. Kerslake, P. Denny, J. Prather. [et al.] //  
arXiv. : официальный сайт. – ISSN 2331-8422. – Текст : электронный.  
– DOI <https://doi.org/10.48550/arXiv.2410.03063>. – URL:  
<https://arxiv.org/abs/2410.03063> (дата обращения: 19.11.2025).

108. Keuning, H. Goodbye Hello World – Research Questions for a Future  
CS1 Curriculum / H. Keuning, A. Luxton-Reilly, A. Petersen [et al.] // In  
Proceedings of the 24th Koli Calling International Conference on Computing

Education Research : сайт. – ISSN отсутствует. – Текст : электронный. – DOI <https://doi.org/10.1145/3699538.3699591>. URL: <https://doi.org/10.1145/3699538.3699591> (дата обращения: 19.04.2025).

109. Kiesler, N. Large Language Models in Introductory Programming Education: ChatGPT's Performance and Implications for Assessments / N. Kiesler, D. Schiffner // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2308.08572.2023>. – URL: <https://arXiv:2308.08572.2023> (дата обращения: 19.11.2025).

110. Kizilcec, R. F. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses / R. F. Kizilcec, C. Piech, E. Schneider // Journal of Learning Analytics. – 2017. – № 4 (3). – P. 41–58. – ISSN 19297750.

111. Koedinger, K. R. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning / K. R. Koedinger, A. T. Corbett, C. Perfetti // Cognitive Science. – 2012. – №36 (5). – P.757–798. – ISSN 0364-0213.

112. Lee, C. From Keyboard to Chatbot: An AI-powered Integration Platform with Large-Language Models for Teaching Computational Thinking for Young Children / C. Lee, J. Xiong // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2405.00750.2024> (дата обращения: 19.11.2025).

113. Lent, R. W. Longitudinal relations of self-efficacy to outcome expectations, interests, and major choice goals in engineering students / R. W. Lent, H. Sheu, D. Singley [et al.] // Journal of Vocational Behavior. – 2008. –№ 73 (2). – P.328–335. – ISSN 0001-8791.

114. Iglesias, C. A. A Survey of Agent-Oriented Methodologies / C. A. Iglesias, M. Garijo, J. C. Gonzalez // Proc of the Intelligent Agents V: Agents Theories, Architectures, and Languages: 5th Int. Workshop, ATAL'98 – Berlin: Springer Berlin Heidelberg, 1999. – P. 317–330.

115. Li, X. Graphteam: Facilitating large language model-based graph analysis via multi-agent collaboration / X. Li, Q. Chu, Y. Chen [et al.] // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2410.18032.2024>. – URL: <https://arXiv:2410.18032.2024> (дата обращения: 19.11.2025).
116. Lu, J. YODA: Teacher-Student Progressive Learning for Language Models / J. Lu, W. Zhong, Y. Wang [et al.] // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2401.15670.2024>. – URL: <https://arXiv:2401.15670.2024> (дата обращения: 19.11.2025).
117. Osborne, F. Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks / F. Osborne, E. Motta // researchgate : сайт. – ISSN отсутствует. – Текст : электронный. – DOI:10.1007/978-3-319-25007-6\_24. – URL: [https://www.researchgate.net/publication/281650037\\_Klink2\\_Integrating\\_Multiple\\_Web\\_Sources\\_to\\_Generate\\_Semantic\\_Topic\\_Networks](https://www.researchgate.net/publication/281650037_Klink2_Integrating_Multiple_Web_Sources_to_Generate_Semantic_Topic_Networks) (дата обращения: 12.04.2025).
118. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) / w3.org : официальный сайт. – Текст : электронный. – URL: <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/> (дата обращения: 19.06.2025).
119. Programming languages: / DBPedia : сайт. – Текст : электронный. – URL: [https://dbpedia.org/describe/?url=http%3A%2F%2Fdbpedia.org%2Fresource%2FCategory%3AProgramming\\_languages&invfp=IFP\\_OFF&sas=SAME\\_AS\\_OFF&sid=13283](https://dbpedia.org/describe/?url=http%3A%2F%2Fdbpedia.org%2Fresource%2FCategory%3AProgramming_languages&invfp=IFP_OFF&sas=SAME_AS_OFF&sid=13283) (дата обращения: 20.06.2025).
120. Pugliese, L. Adaptive Learning Systems: Surviving the Storm / L.Pugliese// educause : сайт. – ISSN отсутствует. – Текст : электронный. – DOI отсутствует. – URL: <https://er.educause.edu/articles/2016/10/adaptive-learning-systems-surviving-the-storm> (дата обращения: 01.06.2024).

121. Qian, C. Experiential co-learning of software-developing agents / C. Qian // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2312.17025.2024>. – URL: <https://arxiv.org/abs/2312.17025.2024> (дата обращения: 19.11.2025).
122. Radmehr, B. Towards Generalizable Agents in Text-Based Educational Environments: A Study of Integrating RL with LLMs / B. Radmehr, A. Singla, T. Käser // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2404.18978.2024>. – URL: <https://arxiv.org/abs/2404.18978.2024> (дата обращения: 19.11.2025).
123. Raudenbush, S. W. Hierarchical Linear Models: Applications and Data Analysis Methods / S. W. Raudenbush, A. S. Bryk // Thousand Oaks: SAGE Publications – 2002. – P. 485-487.
124. Romera-Paredes, B. Mathematical discoveries from program search with large language models / B. Romera-Paredes, M. Barekatain, A. Novikov [et al.] // Nature. – 2024. – № 625 (7995). – С. 468–475. – ISSN 1476-4687. – Текст : электронный. – DOI <https://doi.org/10.1038/s41586-023-06924-6>. (дата обращения: 19.11.2025).
125. Salmon, A. Debugging Without Error Messages: How LLM Prompting Strategy Affects Programming Error Explanation Effectiveness / A. Salmon, K. Hammer, E. A. Santos, B. A. Becker // arXiv. : официальный сайт – ISSN 2331-8422. – Текст : электронный. – DOI отсутствует. – URL: <https://arxiv.org/abs/2501.05706>. 2025 (дата обращения: 19.11.2025).
126. Savelka, J. Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses? / J. Savelka [et al.] // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2303.09325.2023>. – URL: <https://arxiv.org/abs/2303.09325.2023> (дата обращения: 19.11.2025).
127. Singer, J. D Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence / J. D. Singer, J. B. Willett // Oxford University Press. – 2003. – № 3 – P. 644.

128. SKOS Simple Knowledge Organization System / w3.org : официальный сайт. – Текст : электронный. – URL: <https://www.w3.org/TR/skos-reference/#L4858> (дата обращения: 19.06.2023).
129. Snook, I. Invisible Learnings? A Commentary on John Hattie's Visible Learning: A synthesis of over 800 metaanalyses relating to achievement / I. Snook, J. Clark, R. Openshaw [et al.] // Educational Philosophy and Theory. New Zealand Association for Research in Education. – 2009. – Volume 44. P. 555–563. – ISSN 1469-5812. 2199-4714. – Текст : электронный. – DOI <https://doi.org/10.1111/j.1469-5812.2009.00575.x>. – URL <https://scispace.com/papers/invisible-learnings-a-commentary-on-john-hattie-s-book-uxv3xrub99>. (дата обращения: 19.11.2025).
130. Walkington, C. Personalizing algebra to students' individual interests in an intelligent tutoring system: Moderators of impact / C. Walkington, M. Bernacki // International Journal of Artificial Intelligence in Education. – 2019. – 29 (1). P. 58–88. – ISSN 1560-4292.
131. Wang, T. et al. LLM-powered Multi-agent Framework for Goal oriented Learning in Intelligent Tutoring System // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2501.15749.2025>. – URL: <https://arXiv:2501.15749.2025> (дата обращения: 19.11.2025).
132. Xu, S. EduAgent: Generative Student Agents in Learning / S. Xu, X. Zhang, L. Qin // arXiv : официальный сайт – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2404.07963.2024>. – URL: <https://arXiv:2404.07963.2024> (дата обращения: 19.11.2025).
133. Zhang, C. Mathemyths: Leveraging Large Language Models to Teach Mathematical Language through Child-AI Co-Creative Storytelling / C. Zhang [et al.] // arXiv : официальный сайт. – ISSN 2331-8422. – Текст : электронный. – DOI <https://doi.org/10.48550/arXiv.2402.01927.2024>. – URL: <https://arXiv:2402.01927.2024> (дата обращения: 19.11.2025).

134. Zadeh, L. A. Fuzzy Logic, Neural Networks, and Soft Computing / L. A. Zadeh // Communications of the ACM. – 1994. – № 37 (3). – P.77–84. – ISSN 0001-0782.
135. Zhang, W. Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review / W. Zhang, J. Zhang //Mathematics. – 2025. – № 13 (5) – ISSN отсутствует. – Текст : электронный. – DOI отсутствует. – URL: <https://www.mdpi.com/2227-7390/13/5/856> (дата обращения : 22.09.2025).
136. Timofeev, A. N. Development of a Competence Model in the Field of Information Technology Using Ontologies and Knowledge Bases = Разработка модели компетенций в сфере информационных технологий на основе интеграции онтологий и баз знаний / A.N. Timofeev // 2023 16th International Conference Management of large-scale system development (MLSD). – Vienna : IEEE, 2023. – ISBN 979-8-3503-3790-7 – Текст : электронный. – DOI 10.1109/MLSD58227.2023.10303922. – URL: <https://ieeexplore.ieee.org/document/10303922/> (дата обращения: 24.11.2025).
137. Timofeev, A. N. A method based on large language models for prompt adjustment of an individual curriculum during e-practice in large-scale network companies = A method based on large language models for prompt adjustment of an individual curriculum during e-practice in large-scale network companies / S.S. Mikhailova, A.N. Timofeev, A.V. Timoshenko, A.S. Zakharov // 2025 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO). – Vienna : IEEE, 2025. – eISSN 2832-0514 – Текст : электронный. – DOI 10.1109/SYNCHROINFO65403.2025.11079329. – URL: <https://ieeexplore.ieee.org/document/11079329> (дата обращения: 24.11.2025).

**Приложение А**  
**(информационное)**  
**Результаты анализа профессиональных стандартов**

Таблица А.1 – Результаты анализа профессиональных стандартов

Трудовые функции	Системы / инструменты	Языки / нотации	Методологии / техники
1	2	3	4
«Программист»			
Разработка и отладка программного кода			
Формализация и алгоритмизация поставленных задач	Программные продукты для графического отображения алгоритмов	Языки формализации функциональных спецификаций Нотации для графического отображения алгоритмов	-
Написание программного кода с использованием языков программирования, определения и манипулирования данными	Системы управления базами данных Системы управления задачами Среды разработки	Язык программирования Язык манипулирования данными	Методологии разработки программного обеспечения Методологии и технологии проектирования и использования баз данных
Оформление программного кода в соответствии с установленными требованиями	Среды разработки Система контроля версий	-	-
Работа с системой контроля версий	Системы контроля версий	-	-
Проверка и отладка программного кода	Системы статического анализа Системы мониторинга Системы логирования	-	-
Проверка работоспособности и рефакторинг кода программного обеспечения			
Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения	Системы тестирования Системы мониторинга Системы замера характеристик Средства пакетного выполнения	-	-
Разработка тестовых наборов данных	Системы генерации тестовых наборов	-	-
Проверка работоспособности программного обеспечения	Системы управления задачами Системы мониторинга Системы логирования	-	-
Рефакторинг и оптимизация программного кода	Среды разработки	Языки программирования	

Продолжение таблицы А.1

1	2	3	4
Исправление дефектов, зафиксированных в базе данных дефектов	Системы управления задачами/ошибками		Методы и приемы отладки программного кода
<b>Интеграция программных модулей и компонент и верификация выпусков программного продукта</b>			
Разработка процедур интеграции программных модулей	Системы документирования Средства миграции и преобразования данных Средства разработки процедур для развертывания программного обеспечения	Язык программирования	Методы миграции и преобразования данных Методы разработки процедур для развертывания программного обеспечения
Осуществление интеграции программных модулей и компонент и верификации выпусков программного продукта	Система управления жизненным циклом Системы сборки Системы тестирования	-	-
<b>Работка требований и проектирование программного обеспечения</b>			
Анализ требований к программному обеспечению	Система управления требованиями Системы управления задачами	-	Методологии разработки программного обеспечения Методологии и технологии проектирования и использования баз данных
Разработка технических спецификаций на программные компоненты и их взаимодействие	Системы управления задачами Средства проектирования программного обеспечения Средства проектирования программных интерфейсов	Языки формализации функциональных спецификаций	Методы формализации задач Методы проектирования программного обеспечения Методы проектирования программных интерфейсов
Проектирование программного обеспечения	-	Нотации моделирования	-
<b>«Специалист по тестированию в области информационных технологий»</b>			
<b>Подготовка тестовых данных и выполнение тестовых процедур ПО</b>			
Подготовка к выполнению задания на тестирование ПО	Операционные системы Системы тестирования	-	-
Подготовка тестовых данных в соответствии с заданием на тестирование ПО	Генераторы тестовых данных Офисные системы	Форматы представления тестовых данных	Методы работы с базами данных Методы обработки первичной информации/источников первичной информации
Выполнение процесса тестирования ПО	Системы тестирования Системы модульного тестирования системы контроля дефектов ПО	Языки программирования	-

Продолжение таблицы А.1

1	2	3	4
Документирование дефектов ПО	системы контроля дефектов ПО Системы управления задачами Офисные системы	-	-
Тестирование эксплуатационной и технической документации на ПО	Системы управления документацией Офисные системы	-	-
Разработка тестовых случаев, проведение тестирования ПО и исследование результатов	-	-	-
Определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов	Системы тестирования Системы модульного тестирования Системы управления документацией Офисные системы	Языки программирования	Техники проектирования и комбинаторики тестов
Проведение тестирования ПО по разработанным тестовым случаям	Системы тестирования	-	Техники тестирования ПО Стандарты оформления кода
Восстановление работоспособности ПО	Системы автоматизированного тестирования Операционные системы	Языки программирования	Регрессионное тестирование
Анализ результатов тестирования ПО на соответствие ожидаемым результатам, оформление и размещение отчета о тестировании в соответствии с жизненным циклом ПО в системе контроля версий	Системы тестирования Офисные системы Инструменты выполнения тестов	-	Базовые техники проектирования и комбинаторики тестов Метрики тестирования
Проверка устраниенных дефектов ПО в порядке их приоритета	инструменты командной работы	-	Техники тестирования ПО Регрессионное тестирование
Оформление отчета по результатам регрессионного тестирования ПО	Офисные системы	-	Методы оптимизации тестовых наборов
Разработка документов для тестирования ПО и анализ качества тестового покрытия	-	-	-
Верификация требований исходной документации на ПО	Системы управления требованиями	-	Методы анализа и тестирования требований Теория тестирования ПО
Определение требований к тестам	Системы управления требованиями	-	Методы разработки требований Методики анализа рисков Теория тестирования Техники тестирования ПО
Разработка организационных документов для проведения тестирования проекта, включая план тестирования ПО	Системы тестирования	-	Метрики качества Методики анализа рисков Техники тестирования ПО Стандарты качества ПО

Продолжение таблицы А.1

1	2	3	4
			Стандарты по информационной безопасности ПО
Оценка тестов	Системы тестирования	-	Метрики качества Проектовые метрики Методы типовых алгоритмов
Управление процессом тестирования ПО	-	-	-
Выявление приоритетных требований к ПО для покрытия тестами	Системы управления требованиями	-	методы анализа требований проектные методологии Техники и технологии ведения переговоров Жизненный цикл ПО
Согласование требований с заказчиком	-	-	методы анализа требований жизненный цикл ПО, различные методологии его разработки Методы тестирования
Разработка стратегии тестирования ПО	-	-	Методики анализа рисков Теория стратегий тестирования ПО Техники тестирования ПО Стандарты качества ПО
Организация рабочего процесса команды специалистов по тестированию ПО (включая оценку трудозатрат)	инструменты командной работы	-	Жизненный цикл ПО
Мониторинг работ по тестированию ПО и информирование о ходе работ заинтересованных лиц	Основные инструментальные средства организации работы в команде Системы тестирования	-	Методология и методы верификации ПО Методы предотвращения рисков
<p>Примечание – Термины и концепты, используемые в стандартах «Программист» и «Специалист по тестированию в области информационных технологий», можно условно разделить на классы: «Системы или инструменты», «Языки или нотации», «Методы или техники». Рассмотренные классы, понятия и отношения между ними представлены в виде следующего сегмента графа знаний G.</p> <p>Сначала вводится специальное отношение relatedEquivalent, являющееся частным случаем отношения skos:related.</p> <pre>&lt;relatedEquivalent&gt; rdf:type owl:ObjectProperty ; rdfs:subPropertyOf skos:related .</pre> <p>Далее описываются основные классы, относящиеся к профессиональным стандартам и сами стандарты.</p> <pre>&lt;Стандарт&gt; rdf:type owl:Class ; &lt;Обобщенная_трудовая_функция&gt; rdf:type owl:Class ; &lt;Трудовая_функция&gt; rdf:type owl:Class ; Обобщенная_трудовая_функция &lt;skos:broader&gt; &lt;Стандарт&gt; . &lt;Трудовая_функция&gt; &lt;skos:broader&gt; &lt;Обобщенная_трудовая_функция&gt; . &lt;Стандарт_Программист&gt; rdf:type schema:Стандарт ; &lt;Стандарт_Специалист_по_тестированию_в_области_информационных_технологий&gt; rdf:type schema:Стандарт ;</pre>			

Продолжение таблицы А.1

1	2	3	4
	<p>После этого описываются классы для систематизации концептов и терминов, используемых в стандартах.</p> <pre>&lt;Система&gt; rdf:type owl:Class ; &lt;Инструмент&gt; rdf:type owl:Class ; &lt;Язык&gt; rdf:type owl:Class ; &lt;Нотация&gt; rdf:type owl:Class ; &lt;Метод&gt; rdf:type owl:Class ; &lt;Техника&gt; rdf:type owl:Class ; &lt;Система&gt; &lt;relatedEquivalent&gt; &lt;Инструмент&gt; . &lt;Язык&gt; &lt;relatedEquivalent&gt; &lt;Нотация&gt; . &lt;Метод&gt; &lt;relatedEquivalent&gt; &lt;Техника&gt; .</pre> <p>Описывается профессиональный стандарт «Программист» и в качестве примера одна обобщенная трудовая функция и одна трудовая функция, относящиеся к нему.</p> <pre>&lt;Разработка_и_отладка_программного_кода&gt; rdf:type scema:Обобщенная_трудовая_функция ; &lt;Разработка_и_отладка_программного_кода&gt; &lt;skos:broader&gt; &lt;Стандарт_Программист&gt; . &lt;Написание_программного_кода_с_использованием_языков_программирования_определения_и_манипулирования_данными&gt; rdf:type scema: Трудовая_функция ; &lt;Написание_программного_кода_с_использованием_языков_программирования_определения_и_манипулирования_данными&gt; &lt;skos:broader&gt; &lt;Разработка_и_отладка_программного_кода&gt; .</pre> <p>Описывается профессиональный стандарт «Специалист по тестированию в области информационных технологий» и в качестве примера одна обобщенная трудовая функция и одна трудовая функция, относящиеся к нему.</p> <pre>&lt;Подготовка_тестовых_данных_и_выполнение_тестовых_процедур_ПО&gt; rdf:type scema:Обобщенная_трудовая_функция ; &lt;Подготовка_тестовых_данных_и_выполнение_тестовых_процедур_ПО&gt; &lt;skos:broader&gt; &lt;Стандарт_Специалист_по_тестированию_в_области_информационных_технологий&gt; . &lt;Подготовка_тестовых_данных_в_соответствии_с_заданием_на_тестирование_ПО&gt; rdf:type scema: Трудовая_функция ; &lt;Подготовка_тестовых_данных_в_соответствии_с_заданием_на_тестирование_ПО&gt; &lt;skos:broader&gt; &lt;Подготовка_тестовых_данных_и_выполнение_тестовых_процедур_ПО&gt; .</pre> <p>В качестве примера далее описываются отношения между концептами, выделенными в приложении А курсивом, трудовыми функциями и классами концептов.</p> <p>Сначала описываются концепты.</p> <pre>&lt;Система_управления_базами_данных&gt; rdf:type skos:Concept ; &lt;Язык_программирования&gt; rdf:type skos:Concept ; &lt;Методология_разработки_программного_обеспечения&gt; rdf:type skos:Concept ; &lt;Офисная_система&gt; rdf:type skos:Concept ; &lt;Формат_представления_тестовых_данных&gt; rdf:type skos:Concept ; &lt;Метод_работы_с_базами_данных&gt; rdf:type skos:Concept ;</pre> <p>После описания концептов приводятся отношения между концептами, трудовыми функциями и классами концептов.</p> <pre>&lt;Написание_программного_кода_с_использованием_языков_программирования_определения_и_манипулирования_данными&gt; &lt;skos:related&gt; &lt;Система_управления_базами_данных&gt;, &lt;Язык_программирования&gt;, &lt;Методология_разработки_программного_обеспечения&gt; . &lt;Подготовка_тестовых_данных_в_соответствии_с_заданием_на_тестирование_ПО&gt; &lt;skos:related&gt; &lt;Офисная_система&gt;, &lt;Формат_представления_тестовых_данных&gt;, &lt;Метод_работы_с_базами_данных&gt; . &lt;Система&gt; &lt;dcterms:subject&gt; &lt;Системы_управления_базами_данных&gt;, &lt;Офисные_системы&gt; .</pre>		

Продолжение таблицы А.1

1	2	3	4
<p>&lt;Язык&gt; &lt;dcterms:subject&gt; &lt;Язык_программирования&gt;, &lt;Формат_представления_тестовых_данных&gt; . &lt;Метод&gt; &lt;dcterms:subject&gt; &lt;Методология_разработки_программного_обеспечения&gt;, &lt;Метод_работы_с_базами_данных&gt; . Сформированный из профессиональных стандартов граф знаний <math>G</math> связывается с базами знаний <math>Kb^A</math> и <math>Kb^O</math>. Например, концепты &lt;Язык_программирования&gt; и &lt;Системы_управления_базами_данных&gt; будут связаны с соответствующим концептом &lt;Programming languages&gt; и &lt;Database management systems&gt; в <math>Kb^O</math>(DBpedia): &lt;Язык_программирования&gt; &lt;owl:sameAs&gt; &lt;Programming languages&gt; &lt;Системы_управления_базами_данных&gt; &lt;owl:sameAs&gt; &lt;Database management systems&gt;</p>			

Источник: составлено автором.

**Приложение Б**  
**(информационное)**  
**Примеры задач для решения**

Таблица Б.1 – Задачи для решения студентами в рамках курса «Структуры и алгоритмы обработки данных»

Студент	Язык	Задача 1	Задача 2	Задача 3
1	2	3	4	5
1	java	В одномерном массиве, состоящем из N вещественных элементов, вычислите среднее арифметическое элементов массива, расположенных после минимального по модулю элемента.	Ввести строку с клавиатуры. Разделить строку на слова с учетом пробелов и знаков препинания ««.!,?:;»» и вывести каждое слово в отдельной строке. Если за знаком препинания или пробелом сразу идет другой знак препинания или пробел, то между ними слова нет.	Дан список объектов класса «Квартира»: ФИО владельца, площадь, число комнат, этаж. Для данного списка реализовать следующие операции, выполняемые с помощью консольного ввода: 1) Создание нового элемента списка объектов (ввод его данных); 2) Сортировка списка объектов (сортировать в алфавитном порядке по ФИО, входящему в класс, использовать готовые/встроенные реализации алгоритма сортировки запрещено); 3) Поиск в списке объектов по заданному параметру (самостоятельно выберите поле структуры и реализуйте поиск по нему), вывод желательно сделать в таблице (как в функции под номером 6); 4) Изменение заданного элемента списка(можно сделать так: вводится

Продолжение таблицы Б.1

1	2	3	4	5
				номер элемента в списке объектов, и его данные (значения элементов структуры) изменяются на новые введенные пользователем данные); 5) Удаление объекта из списка; 6) Вывод на экран списка объектов в виде таблицы. Также выводите количество элементов в списке(или номеруйте выводимые строки).
2	python	<p>Дан список объектов класса «Сотрудник»: ФИО; должность; год рождения; заработка плата. Для данного списка реализовать следующие операции, выполняемые с помощью консольного ввода:</p> <p>1) Создание нового элемента списка объектов (ввод его данных);      2) Сортировка списка объектов (сортировать по зарплате, использовать готовые/встроенные реализации алгоритма сортировки запрещено);      3) Поиск в списке объектов по заданному параметру (самостоятельно выберите поле структуры и реализуйте поиск по нему), вывод желательно сделать в таблице (как в функции под номером 6);      4) Изменение заданного элемента списка(можно сделать так: вводится</p>	<p>Дана целочисленная матрица A (N, N). Поменяйте местами элементы главной и побочной диагоналей.</p>	<p>Ввести строку с клавиатуры. Разделить строку на слова с учетом пробелов и знаков препинания «.,!,:» и вывести каждое слово в отдельной строке. Если за знаком препинания или пробелом сразу идет другой знак препинания или пробел, то между ними слова нет.</p>

Продолжение таблицы Б.1

1	2	3	4	5
		<p>номер элемента в списке объектов, и его данные (значения элементов структуры) изменяются на новые введенные пользователем данные);</p> <p>5) Удаление объекта из списка;</p> <p>6) Вывод на экран списка объектов в виде таблицы. Также выводите количество элементов в списке (или номеруйте выводимые строки).</p>		
3	java	<p>Ввести две строки с клавиатуры. Сформировать третью строку таким образом, чтобы в нее попаременно вошли четные символы первой строки и нечетные символы второй строки. В качестве длины третьей строки взять длину меньшей из введенных строк.</p>	<p>Дан список объектов класса «Квартира»: ФИО владельца, площадь, число комнат, этаж. Для данного списка реализовать следующие операции, выполняемые с помощью консольного ввода:</p> <p>1) Создание нового элемента списка объектов (ввод его данных);</p> <p>2) Сортировка списка объектов (сортировать в алфавитном порядке по ФИО, входящему в класс, использовать готовые/встроенные реализации алгоритма сортировки запрещено);</p> <p>3) Поиск в списке объектов по заданному параметру (самостоятельно выберите поле структуры и реализуйте поиск по нему), вывод желательно сделать в таблице (как в функции под номером 6);</p> <p>4) Изменение заданного элемента списка(можно сделать так: вводится</p>	<p>Дан одномерный массив строк. Отсортируйте его методом пузырька</p>

Продолжение таблицы Б.1

1	2	3	4	5
			<p>номер элемента в списке объектов, и его данные (значения элементов структуры) изменяются на новые введенные пользователем данные);</p> <p>5) Удаление объекта из списка;</p> <p>6) Вывод на экран списка объектов в виде таблицы. Также выводите количество элементов в списке(или номеруйте выводимые строки).</p>	
4	python	Дан одномерный массив строк. Отсортируйте его методом пузырька	<p>Ввести две строки с клавиатуры.</p> <p>Сформировать третью строку таким образом, чтобы в нее попеременно вошли четные символы первой строки и нечетные символы второй строки. В качестве длины третьей строки взять длину меньшей из введенных строк.</p>	<p>Дан список объектов класса «Сотрудник»: ФИО; должность; год рождения; заработка плата. Для данного списка реализовать следующие операции, выполняемые с помощью консольного ввода:</p> <p>1) Создание нового элемента списка объектов (ввод его данных);</p> <p>2) Сортировка списка объектов (сортировать по зарплате, использовать готовые/встроенные реализации алгоритма сортировки запрещено);</p> <p>3) Поиск в списке объектов по заданному параметру (самостоятельно выберите поле структуры и реализуйте поиск по нему), вывод желательно сделать в таблице (как в функции под номером 6);</p> <p>4) Изменение заданного элемента</p>

Продолжение таблицы Б.1

1	2	3	4	5
				списка(можно сделать так: вводится номер элемента в списке объектов, и его данные (значения элементов структуры) изменяются на новые введенные пользователем данные); 5) Удаление объекта из списка; 6) Вывод на экран списка объектов в виде таблицы. Также выводите количество элементов в списке(или номеруйте выводимые строки).
5	python	Ввести строку с клавиатуры. Определить является ли строка правильно введенной датой. Введенные числа должны подходить под ограничения количества дней и месяцев. Проверьте следующие форматы дат: dd/mm/yyyy, dd.mm.yyyy.	Дан одномерный массив чисел. Отсортируйте его методом Шелла	"Дан список объектов класса «Государство»: название; столица; численность населения; занимаемая площадь. Для данного списка реализовать следующие операции, выполняемые с помощью консольного ввода: 1) Создание нового элемента списка объектов (ввод его данных); 2) Сортировка списка объектов (сортировать по плотности населения (численность населения/занимаемая площадь), использовать готовые/встроенные реализации алгоритма сортировки запрещено); 3) Поиск в списке объектов по заданному параметру (самостоятельно выберите поле структуры и реализуйте поиск по нему), вывод желательно

Продолжение таблицы Б.1

1	2	3	4	5
				<p>сделать в таблице (как в функции под номером 6);</p> <p>4) Изменение заданного элемента списка(можно сделать так: вводится номер элемента в списке объектов, и его данные (значения элементов структуры) изменяются на новые введенные пользователем данные);</p> <p>5) Удаление объекта из списка;</p> <p>6) Вывод на экран списка объектов в виде таблицы. Также выводите количество элементов в списке (или номеруйте выводимые строки)."</p>
6	python	Дана целочисленная матрица A (N, N). Поменяйте местами элементы главной и побочной диагоналей.	Ввести строку с клавиатуры. Определить является ли строка правильно введенной датой. Введенные числа должны подходить под ограничения количества дней и месяцев. Проверьте следующие форматы дат: dd/mm/yyyy, dd.mm.yyyy.	Дан одномерный массив чисел. Отсортируйте его методом Шелла
7	python	Ввести строку и число с клавиатуры. Сдвинуть строку на введенное число символов, но пробелы оставить на месте. К примеру, строка «Я люблю программировать» и число 2 преобразовывается в строку «т Ёлюб люпрограммирова».	"Дан список объектов класса «Государство»: название; столица; численность населения; занимаемая площадь. Для данного списка реализовать следующие операции, выполняемые с помощью консольного ввода:	В одномерном массиве, состоящем из N вещественных элементов, вычислите среднее арифметическое элементов массива, расположенных после минимального по модулю элемента.

Продолжение таблицы Б.1

1	2	3	4	5
			<p>(сортировать по плотности населения (численность населения/занимаемая площадь), использовать готовые/встроенные реализации алгоритма сортировки запрещено);</p> <p>3) Поиск в списке объектов по заданному параметру (самостоятельно выберите поле структуры и реализуйте поиск по нему), вывод желательно сделать в таблице (как в функции под номером 6);</p> <p>4) Изменение заданного элемента списка(можно сделать так: вводится номер элемента в списке объектов, и его данные (значения элементов структуры) изменяются на новые введенные пользователем данные);</p> <p>5) Удаление объекта из списка;</p> <p>6) Вывод на экран списка объектов в виде таблицы. Также выводите количество элементов в списке (или номеруйте выводимые строки)."</p>	222

Источник: составлено автором.

**Приложение В**  
**(информационное)**  
**Сценарии взаимодействия**

Таблица В.1 – Сценарии взаимодействия обучающегося и НГО-агента

Диалог 1	
Пользователь (Вы)	Агент – разработчик
-	Перед началом диалога агенту задается контекст. Своими словами пишите, что агент – это разработчик программного обеспечения, обладающий опытом в определенном языке программирования. Контекст задается 4-5 предложениями
-	Агент ответил, что понял контекст
Пишите, что вам необходимо помочь с решением задачи. Указываете условие задачи. Просите привести решение на заданном языке программирования	-
-	Агент должен вернуть вам код
Уточняете, что код должен подходить под структуру функции, которую вы реализовали вручную. Описываете структуру этой функции то есть выходные и выходные параметры, ее имя.	-
-	Агент должен вернуть вам код, подходящий под структуру функции. Если он не смог сделать это с первого раза, то просите до того, момента пока код не подходит под структуру
Полученный от агента код вставляете в свою функцию и запускаете с вашим тестовым примером	-
Если код выдал результаты НЕ совпадающие с тестовым примером, то переходите к строке 8 иначе переходите к строке 12	-

Продолжение таблицы В.1

Пишите агенту, что запустили его код с входными данными (взятыми из вашего теста) и ожидаете выходные данные (взятые из вашего теста). Просите агента исправить код и написать, какие ошибки он исправил	Агент должен вернуть вам код и описание исправленных ошибок
Переходите к строке 8	
Просите агента объяснить, почему он привел именно такое решение	Агент должен вернуть вам объяснение
Диалог 2	
Пользователь (Вы)	Агент – разработчик
-	Перед началом диалога агенту задается контекст. Своими словами пишите, что агент – это разработчик программного обеспечения, обладающий опытом в определенном языке программирования. Контекст задается 4-5 предложениями
-	Агент ответил, что понял контекст
Пишите, что вы решили задачу (указываете условие задания) путем написания следующего кода (указываете код из вашего эталонного примера). Просите агента исправить этот код и пояснить, почему он внес именно такие исправления	-
-	Агент должен вернуть вам код и описание исправленных проблем
Полученный от агента код вставляете в свою функцию и запускаете с вашим тестовым примером	-
Если код выдал результаты НЕ совпадающие с тестовым примером, то переходите к строке 8 иначе переходите к строке 10	-
Пишите агенту, что запустили его код с входными данными (взятыми из вашего теста) и ожидаете выходные данные (взятые из вашего теста). Просите агента исправить код и написать, какие ошибки он исправил	Агент должен вернуть вам код и описание исправленных ошибок
Переходите к строке 6	-
Просите агента объяснить, почему он привел именно такое решение	Агент должен вернуть вам объяснение

Продолжение таблицы В.1

Диалог 3	
Пользователь (Вы)	Агент – разработчик
-	Перед началом диалога агенту задается контекст. Своими словами пишите, что агент – это разработчик программного обеспечения, обладающий опытом в определенном языке программирования. Контекст задается 4-5 предложениями
-	Агент ответил, что понял контекст
Пишите, что вы решили задачу (указываете условие задания) путем написания следующего кода (указываете код из вашего эталонного примера). Просите агента дать советы по исправлению этого кода	-
-	Агент должен вернуть вам перечень советов по исправлению
По советам агента исправляете код. Если совет не понятен, своими словами просите уточнить агента, до тех пор, пока у вас не получится внести исправление в ваш код	-
Запускаете исправленный код на тестовых примерах	-
Если код выдал результаты НЕ совпадающие с тестовым примером, то переходите к строке 9 иначе переходите к строке 11	-
Пишите агенту, что по его советам исправили ваш код, запустили на нем тесты (указываете входные и выходные данные), но тесты не проходят. Приводите исправленный вами код. Просите агента объяснить, почему тесты не проходят и какой из его советов был неправильным	Агент должен вернуть вам перечень советов по исправлению
Переходите к строке 6	-
Просите агента объяснить, почему он привел именно такое решение	Агент должен вернуть вам объяснение
Диалог 4	
Пользователь (Вы)	Агент – тестировщик
-	Перед началом диалога агенту задается контекст. Своими словами пишите, что агент – это тестировщик программного

Продолжение таблицы В.1

	обеспечения, обладающий опытом тестирования ПО. Контекст задается 4-5 предложениями
-	Агент ответил, что понял контекст
Пишите агенту условие данной вам задачи	-
-	Агент ответил, что понял условие
Пишите агенту, что вы решили задачу, приводите код решения данной задачи. Пишите, что для кода вы сделали тестовый пример. Просите агента дать советы по улучшению тестового примера, надо ли в нем что-то исправлять	-
-	Агент должен вернуть вам перечень советов и возможных исправлений
По советам агента исправляете тесты или код. Если совет не понятен, своими словами просите уточнить агента, до тех пор, пока у вас не получится внести исправление в ваш код	-
Выполняете Диалог 2 с исправленными тестами и/или кодом	-
Диалог 5	
Пользователь (Вы)	Агент – тестировщик
-	Перед началом диалога агенту задается контекст. Своими словами пишите, что агент – это тестировщик программного обеспечения, обладающий опытом тестирования ПО. Контекст задается 4-5 предложениями
-	Агент ответил, что понял контекст
Пишите агенту условие данной вам задачи	-
-	Агент ответил, что понял контекст
Пишите агенту, что вы решили задачу, приводите код решения данной задачи. Пишите, что для кода вы сделали тестовый пример. Просите агента сгенерировать тестовый пример по аналогии	-

Продолжение таблицы В.1

	Агент должен вернуть вам новый тестовый пример
Выполняете Диалог 2 с исправленными тестами и/или кодом	-
Диалог 6	
Пользователь (Вы)	Агент – аналитик
-	Перед началом диалога агенту задается контекст. Своими словами пишите, что агент – это аналитик. Контекст задается 4-5 предложениями
-	Агент ответил, что понял контекст
Пишите агенту условие данной вам задачи. Просите его уточнить, что именно агенту не понятно в условиях данной задачи	-
-	Агент вернул вам перечень своих вопросов
Своими словами отвечаете агенту на вопросы. Уточняете, понятны ли агенту ответы. Если не понятны, то пишите пояснение еще раз, если понятны, то переходите к строке 8	-
С учетом ваших пояснений и исходного условия задачи просите агента изложить задачу	-
-	Агент возвращает текст задачи со своими уточнениями
Выполняете Диалог 1 с условием задач, которое изложил агент	-

Источник: составлено автором.

**Приложение Г**  
**(информационное)**  
**Пример диалога обучающегося с агентом**

**Диалог 1**

Контекст: ты – опытный разработчик программного обеспечения, специализирующийся на языке программирования Java. Ты имеешь глубокие знания в области разработки программного обеспечения, а также опыт работы с базами данных и машинным обучением. Также обладаешь навыками написания эффективного и чистого кода, следуя лучшим практикам разработки и постоянно совершенствуешь свои навыки через изучение новых технологий и участие в профессиональных сообществах.

**Ответ**

Звучит интересно! Я готов поделиться своими знаниями и опытом в разработке программного обеспечения на Java. Если у вас есть какие-то конкретные вопросы или задачи, я буду рад помочь.

Задача следующая: Ввести две строки с клавиатуры. Сформировать третью строку таким образом, чтобы в нее попеременно вошли четные символы первой строки и нечетные символы второй строки. В качестве длины третьей строки взять длину меньшей из введенных строк. Необходимо написать код по ней на языке Java.

**Ответ**

Вот пример кода на Java, который решает поставленную задачу:

```
import java.util.Scanner;

public class StringShuffler {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Чтение двух строк с клавиатуры
        String firstString = scanner.nextLine();
        String secondString = scanner.nextLine();

        // Длина меньшей из строк будет длиной результирующей строки
        int resultLength = Math.min(firstString.length(), secondString.length());

        // Создание результирующей строки
        StringBuilder result = new StringBuilder();

        // Перебор символов обеих строк
        for (int i = 0; i < resultLength; i++) {
            // Добавление четного символа из первой строки, если текущий индекс
            четный
            if (i % 2 == 0) {
                result.append(firstString.charAt(i));
            } else { // Добавление нечетного символа из второй строки, если индекс
            нечетный
                result.append(secondString.charAt(i));
            }
        }
    }
}
```

```

    }

    // Вывод результата
    System.out.println(result.toString());
}

}

```

Этот код сначала считывает две строки с клавиатуры, затем определяет длину меньшей из них и создает новую строку, добавляя в неё символы из исходных строк в соответствии с условием задачи. В конце результат выводится на экран.

Код должен подходить под данную структуру функции:

```

public static String
SplitUpAndMerge (String str1, String str2){      int min = Math.min(str1.length(),str2.length());
String s3 = "";    for (int i = 0; i < min; i++){      if (i % 2 == 0){          char tmp = str1.charAt(i);
s3 = s3 + tmp;      }      else {          char tmp = str2.charAt(i);          s3 = s3 + tmp;      }
}
return s3;  }

```

### Ответ

Ваш код уже почти идеален, но он может быть немного оптимизирован для лучшего понимания и поддержки. Вот обновленный вариант:

```

public static String splitUpAndMerge(String str1, String str2) {
    int min = Math.min(str1.length(), str2.length());
    String s3 = "";

    // Используем цикл for для перебора символов в каждой строке
    for (int i = 0; i < min; i++) {
        // Проверяем, является ли индекс четным, чтобы добавить символ из str1
        if (i % 2 == 0) {
            s3 += str1.charAt(i);
        } else { // Если индекс нечетный, добавляем символ из str2
            s3 += str2.charAt(i);
        }
    }

    return s3;
}

```

Изменения, которые я внес:

1. Изменил имя функции на `splitUpAndMerge`, так как `SplitUpAndMerge` не соответствует соглашению об именовании методов в Java (имена методов обычно начинаются с нижнего регистра).
2. Убрал пробелы вокруг операции сложения (`+=`), чтобы сделать код более читаемым.
3. Исправил комментарии в коде, чтобы они соответствовали синтаксису Java.
4. Использовал оператор `+=`, который более читаем и эффективен при добавлении символов к строке.

Твой код был запущен с следующими тестовыми данными:  
`str1 = "Тестовая"`  
`str2 = "Строка"`

Ожидается ответ: Ттсооайспользуй их и исправь код в случае ошибок.

### Ответ

К сожалению, предоставленный вами код не работает корректно с данными тестовыми значениями. После исправления ошибок и оптимизации, вот рабочий вариант функции:

```
public static String splitUpAndMerge(String str1, String str2) {
    int min = Math.min(str1.length(), str2.length());
    String s3 = "";

    // Используем цикл for для перебора символов в каждой строке
    for (int i = 0; i < min; i++) {
        // Проверяем, является ли индекс четным, чтобы добавить символ из str1
        if (i % 2 == 0) {
            s3 += str1.charAt(i);
        } else { // Если индекс нечетный, добавляем символ из str2
            s3 += str2.charAt(i);
        }
    }

    return s3;
}
```

Когда мы запускаем этот код с тестовыми данными `str1 = "Тестовая"` и `str2 = "Строка"`, результатом будет `"Ттсооа"`, как и ожидалось.

Почему ты решил задачу именно так?

### Ответ

Я решил эту задачу таким образом, потому что задача требует создать новую строку, которая состоит из четных символов одной строки и нечетных символов другой строки. Для этого я использовал цикл `for`, который проходит по всем символам обеих строк до тех пор, пока не достигнет минимальной длины этих строк. Внутри цикла я проверяю, является ли индекс текущего символа четным или нечетным, и в зависимости от этого добавляю символ либо из первой строки, либо из второй строки в новую строку `s3`.

**Приложение Д**  
**(информационное)**

**Вычислительный эксперимент. Первая группа**

Таблица Д.1 – Вычислительный эксперимент. Первая группа

Студент	Тема	Тема	Этап	Сложность практики	Факт t^CPC (ч)	Факт t^Kont (ч)	Факт t^CPPI (ч)	Ctrle (баллы)	Rtr (баллы)
1	2	3	4	5	6	7	8	9	10
1	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	3	5,5	7	3,55	3,55
1	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	2	0,5	1,5	3,475	7,025
1	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	простая	5	5	4,5	3,825	10,85
1	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2	4,5	5,5	3,775	14,625
1	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1,5	0,5	1	3,775	18,4
1	2	Двумерные массивы и алгоритмы обработки матриц	Практика	средняя	5	4,5	7	3,5	21,9
1	3	Алгоритмы сортировки	Теория	общая	0,5	2,5	7,5	4,3	26,2
1	3	Алгоритмы сортировки	Опрос	общая	2	0,5	3,5	3,5	29,7
1	3	Алгоритмы сортировки	Практика	средняя	5,5	9,5	7	3,075	32,775
1	4	Алгоритмы поиска	Теория	общая	2	2,5	6,5	4,2	36,975
1	4	Алгоритмы поиска	Опрос	общая	1	0,5	1,5	4,025	41
1	4	Алгоритмы поиска	Практика	средняя	5	5	5	3,45	44,45
1	5	Линейные структуры данных: стек и очередь	Теория	общая	3	5,5	7	3,3	47,75
1	5	Линейные структуры данных: стек и очередь	Опрос	общая	2	0,5	2	3,4	51,15
1	5	Линейные структуры данных: стек и очередь	Практика	простая	6	4,5	2,5	3,675	54,825
1	6	Структуры данных и пользовательские типы	Теория	общая	3	6	5,5	3,925	58,75
1	6	Структуры данных и пользовательские типы	Опрос	общая	2	0,5	2	3,95	62,7
1	6	Структуры данных и пользовательские типы	Практика	сложная	3,5	7	9	3,625	66,325
2	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	2,5	4	5,5	3,875	3,875

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
2	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	2,5	0,5	1,5	3,5	7,375
2	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	слож-ная	6	4,5	7	3,15	10,525
2	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2,5	3,5	7	3,675	14,2
2	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	2	0,5	1,5	3,35	17,55
2	2	Двумерные массивы и алгоритмы обработки матриц	Практика	прос-тая	6	5	5	3,85	21,4
2	3	Алгоритмы сортировки	Теория	общая	2	7	12,5	3,625	25,025
2	3	Алгоритмы сортировки	Опрос	общая	1	0,5	2,5	3,925	28,95
2	3	Алгоритмы сортировки	Практика	сред-няя	8	7,5	6,5	3,3	32,25
2	4	Алгоритмы поиска	Теория	общая	2,5	5,5	4	3,575	35,825
2	4	Алгоритмы поиска	Опрос	общая	1,5	1	1	3,5	39,325
2	4	Алгоритмы поиска	Практика	сред-няя	4	6,5	4,5	3,15	42,475
2	5	Линейные структуры данных: стек и очередь	Теория	общая	1	6	8	3,4	45,875
2	5	Линейные структуры данных: стек и очередь	Опрос	общая	1,5	0,5	1,5	3,95	49,825
2	5	Линейные структуры данных: стек и очередь	Практика	слож-ная	9	6	5	3,175	53
2	6	Структуры данных и пользовательские типы	Теория	общая	3,5	5	7,5	3,35	56,35
2	6	Структуры данных и пользовательские типы	Опрос	общая	2	0,5	1	3,9	60,25
2	6	Структуры данных и пользовательские типы	Практика	сред-няя	8	6	8,5	3,55	63,8
3	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	4	1	7	3,925	3,925
3	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	1,5	0,5	0,5	3,65	7,575
3	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	прос-тая	3	9,5	5	3,625	11,2
3	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2,5	3,5	4,5	4,25	15,45
3	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	2	0,5	2	3,4	18,85
3	2	Двумерные массивы и алгоритмы обработки матриц	Практика	слож-ная	5,5	8	6,5	2,75	21,6
3	3	Алгоритмы сортировки	Теория		3	6,5	12,5	3,3	24,9
3	3	Алгоритмы сортировки	Опрос		1,5	0,5	1,5	3,9	28,8

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
3	3	Алгоритмы сортировки	Прак-тика	сред-няя	7,5	7	6	3,15	31,95
3	4	Алгоритмы поиска	Теория	общая	0,5	6	4,5	3,725	35,675
3	4	Алгоритмы поиска	Опрос	общая	1,5	0,5	1,5	3,825	39,5
3	4	Алгоритмы поиска	Прак-тика	слож-ная	5,5	5,5	7,5	3,325	42,825
3	5	Линейные структуры данных: стек и очередь	Теория	общая	2,5	6	6	3,3	46,125
3	5	Линейные структуры данных: стек и очередь	Опрос	общая	2	0,5	1,5	3,6	49,725
3	5	Линейные структуры данных: стек и очередь	Прак-тика	сред-няя	5,5	1,5	6	3,9	53,625
3	6	Структуры данных и пользовательские типы	Теория	общая	1,5	5,5	5,5	4,25	57,875
3	6	Структуры данных и пользовательские типы	Опрос	общая	0,5	0,5	2,5	4,05	61,925
3	6	Структуры данных и пользовательские типы	Прак-тика	слож-ная	2,5	5,5	7	3,8	65,725
4	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	2	4,5	9,5	3,6	3,6
4	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	1	0,5	1	3,975	7,575
4	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Прак-тика	сред-няя	2,5	5,5	5,5	3,475	11,05
4	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2,5	5	3,5	3,825	14,875
4	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	0,5	0,5	1,5	3,9	18,775
4	2	Двумерные массивы и алгоритмы обработки матриц	Прак-тика	сред-няя	4	4	5,5	3,65	22,425
4	3	Алгоритмы сортировки	Теория	общая	4,5	2,5	8	4,025	26,45
4	3	Алгоритмы сортировки	Опрос	общая	2,5	0	2	3,9	30,35
4	3	Алгоритмы сортировки	Прак-тика	сред-няя	3,5	8	6,5	3,75	34,1
4	4	Алгоритмы поиска	Теория	общая	3	3	5	3,975	38,075
4	4	Алгоритмы поиска	Опрос	общая	2	0,5	2	3,4	41,475
4	4	Алгоритмы поиска	Прак-тика	сред-няя	4	4	4	4,35	45,825
4	5	Линейные структуры данных: стек и очередь	Теория	общая	2,5	4,5	10	3,25	49,075
4	5	Линейные структуры данных: стек и очередь	Опрос	общая	1,5	0,5	1	3,9	52,975
4	5	Линейные структуры данных: стек и очередь	Прак-тика	слож-ная	5,5	6,5	3,5	3,175	56,15
4	6	Структуры данных и пользовательские типы	Теория	общая	2,5	6	10	3,7	59,85

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
4	6	Структуры данных и пользовательские типы	Опрос	общая	2,5	0,5	2	3,5	63,35
4	6	Структуры данных и пользовательские типы	Практика	простая	3,5	7,5	4	3,775	67,125
5	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	2	2,5	6	4,075	4,075
5	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	2	0,5	1,5	3,35	7,425
5	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	простая	5,5	7	5	3,2	10,625
5	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2	6,5	6,5	3,65	14,275
5	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1,5	0,5	1	3,775	18,05
5	2	Двумерные массивы и алгоритмы обработки матриц	Практика	средняя	4	6	5	3,35	21,4
5	3	Алгоритмы сортировки	Теория	общая	3,5	7	4	3,725	25,125
5	3	Алгоритмы сортировки	Опрос	общая	1,5	0,5	1	4	29,125
5	3	Алгоритмы сортировки	Практика	сложная	6,5	6	4,5	3,675	32,8
5	4	Алгоритмы поиска	Теория	общая	1,5	2,5	6,5	4,3	37,1
5	4	Алгоритмы поиска	Опрос	общая	2	0,5	0,5	3,55	40,65
5	4	Алгоритмы поиска	Практика	простая	3	4,5	4,5	3,9	44,55
5	5	Линейные структуры данных: стек и очередь	Теория	общая	2,5	4	8	4,05	48,6
5	5	Линейные структуры данных: стек и очередь	Опрос	общая	2	0,5	1,5	3,6	52,2
5	5	Линейные структуры данных: стек и очередь	Практика	простая	6,5	5,5	8	3,1	55,3
5	6	Структуры данных и пользовательские типы	Теория	общая	4	7	6,5	3,4	58,7
5	6	Структуры данных и пользовательские типы	Опрос	общая	2,5	0,5	2	3,625	62,325
5	6	Структуры данных и пользовательские типы	Практика	сложная	7,5	6	6,5	3,6	65,925
6	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	3,5	7	1,5	3,3	3,3
6	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	1	0,5	2	3,7	7
6	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	сложная	6,5	7,5	6	3,25	10,25
6	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2,5	3	4,5	4,45	14,7

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
6	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1,5	0,5	2	3,625	18,325
6	2	Двумерные массивы и алгоритмы обработки матриц	Практика	слож-ная	5,5	7,5	4	3,3	21,625
6	3	Алгоритмы сортировки	Теория	общая	3	6	4	4,025	25,65
6	3	Алгоритмы сортировки	Опрос	общая	2	0,5	2,5	3,35	29
6	3	Алгоритмы сортировки	Практика	прос-тая	6	6	7	3,725	32,725
6	4	Алгоритмы поиска	Теория	общая	3,5	4	6	3,7	36,425
6	4	Алгоритмы поиска	Опрос	общая	2	0,5	1	3,925	40,35
6	4	Алгоритмы поиска	Практика	сред-няя	4,5	5	5,5	3,6	43,95
6	5	Линейные структуры данных: стеки и очередь	Теория	общая	2,5	5,5	6,5	3,75	47,7
6	5	Линейные структуры данных: стеки и очередь	Опрос	общая	2	0,5	1,5	3,475	51,175
6	5	Линейные структуры данных: стеки и очередь	Практика	сред-няя	2,5	2,5	4,5	4,425	55,6
6	6	Структуры данных и пользовательские типы	Теория	общая	2,5	6	10	3,7	59,3
6	6	Структуры данных и пользовательские типы	Опрос	общая	2	0,5	1,5	3,675	62,975
6	6	Структуры данных и пользовательские типы	Практика	прос-тая	2	6	6	4,025	67
7	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	2,5	5	5,5	3,6	3,6
7	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	1,5	0,5	1,5	3,825	7,425
7	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	сред-няя	4,5	4	5,5	3,675	11,1
7	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2,5	3	6	4,1	15,2
7	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1	0,5	0,5	4,225	19,425
7	2	Двумерные массивы и алгоритмы обработки матриц	Практика	сред-няя	2	4,5	3	3,85	23,275
7	3	Алгоритмы сортировки	Теория	общая	3	5	2	4,15	27,425
7	3	Алгоритмы сортировки	Опрос	общая	2	0,5	1	3,775	31,2
7	3	Алгоритмы сортировки	Практика	сред-ния	6,5	4	4,5	4,2	35,4
7	4	Алгоритмы поиска	Теория	общая	2	4,5	8	3,575	38,975
7	4	Алгоритмы поиска	Опрос	общая	1,5	0,5	1,5	3,95	42,925
7	4	Алгоритмы поиска	Практика	слож-ная	5,5	7,5	3	3,275	46,2
7	5	Линейные структуры данных: стеки и очередь	Теория	общая	2,5	3,5	10	3,7	49,9

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
7	5	Линейные структуры данных: стек и очередь	Опрос	общая	1	0,5	2	3,45	53,35
7	5	Линейные структуры данных: стек и очередь	Практика	простая	5,5	4,5	5,5	3,5	56,85
7	6	Структуры данных и пользовательские типы	Теория	общая	2,5	6	6,5	3,9	60,75
7	6	Структуры данных и пользовательские типы	Опрос	общая	1	0,5	2	3,9	64,65
7	6	Структуры данных и пользовательские типы	Практика	сложная	8,5	4	5	3,75	68,4
7	6	Структуры данных и пользовательские типы	Доп. теория		1	4,5	7,5	4,35	72,75
7	6	Структуры данных и пользовательские типы	Доп. опрос		2,5	0,5	2	3,25	76
7	6	Структуры данных и пользовательские типы	Доп. практика		8,5	5	5,5	3,45	79,45
8	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	1	2,5	3,5	4,625	4,625
8	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	0,5	0,5	1	4,1	8,725
8	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	средняя	6,5	7	6	3	11,725
8	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	3	6,5	5	3,425	15,15
8	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1,5	0,5	1	3,9	19,05
8	2	Двумерные массивы и алгоритмы обработки матриц	Практика	средняя	5,5	5,5	2,5	3,6	22,65
8	3	Алгоритмы сортировки	Теория	общая	3	5	7,5	3,825	26,475
8	3	Алгоритмы сортировки	Опрос	общая	1	0,5	1,5	3,875	30,35
8	3	Алгоритмы сортировки	Практика	средняя	6	8,5	5	3,275	33,625
8	4	Алгоритмы поиска	Теория	общая	3,5	3,5	1,5	3,65	37,275
8	4	Алгоритмы поиска	Опрос	общая	2	0,5	1	3,8	41,075
8	4	Алгоритмы поиска	Практика	сложная	5,5	5,5	7	3,075	44,15
8	5	Линейные структуры данных: стек и очередь	Теория	общая	2	4,5	4	4,125	48,275
8	5	Линейные структуры данных: стек и очередь	Опрос	общая	1	0,5	2,5	3,825	52,1
8	5	Линейные структуры данных: стек и очередь	Практика	сложная	6,5	9,5	6,5	2,775	54,875
8	6	Структуры данных и пользовательские типы	Теория	общая	3	4,5	10,5	3,575	58,45
8	6	Структуры данных и пользовательские типы	Опрос	общая	3	0,5	2	3,25	61,7

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
8	6	Структуры данных и пользовательские типы	Практика	простая	6,5	5,5	8	3,625	65,325
9	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	2	5	3	4,05	4,05
9	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	2	0,5	1,5	3,85	7,9
9	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	сложная	5	4	2,5	4,1	12
9	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2,5	8	8	3,325	15,325
9	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1,5	0,5	1,5	3,825	19,15
9	2	Двумерные массивы и алгоритмы обработки матриц	Практика	средняя	4,5	5,5	2,5	3,7	22,85
9	3	Алгоритмы сортировки	Теория	общая	3	5	9,5	3,6	26,45
9	3	Алгоритмы сортировки	Опрос	общая	2,5	0,5	2,5	3,525	29,975
9	3	Алгоритмы сортировки	Практика	сложная	3,5	5,5	4	3,975	33,95
9	4	Алгоритмы поиска	Теория	общая	2,5	5	6,5	3,5	37,45
9	4	Алгоритмы поиска	Опрос	общая	1,5	0,5	1,5	3,575	41,025
9	4	Алгоритмы поиска	Практика	средняя	3,5	4,5	4	3,775	44,8
9	5	Линейные структуры данных: стек и очередь	Теория	общая	1,5	3	5,5	4,15	48,95
9	5	Линейные структуры данных: стек и очередь	Опрос	общая	1,5	0,5	1	3,9	52,85
9	5	Линейные структуры данных: стек и очередь	Практика	сложная	7	4	5,5	3,475	56,325
9	6	Структуры данных и пользовательские типы	Теория	общая	3,5	6	4,5	3,675	60
9	6	Структуры данных и пользовательские типы	Опрос	общая	2	0,5	1,5	3,55	63,55
9	6	Структуры данных и пользовательские типы	Практика	простая	5,5	6,5	5,5	3,7	67,25
10	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	общая	1,5	3,5	5	4,075	4,075
10	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	общая	1,5	0,5	2	3,625	7,7
10	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	средняя	5,5	4,5	4	3,575	11,275
10	2	Двумерные массивы и алгоритмы обработки матриц	Теория	общая	2	4,5	4,5	3,9	15,175
10	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	общая	1,5	0	1,5	4,225	19,4

Продолжение таблицы Д.1

1	2	3	4	5	6	7	8	9	10
10	2	Двумерные массивы и алгоритмы обработки матриц	Практика	средняя	3,5	3,5	4	4,1	23,5
10	3	Алгоритмы сортировки	Теория	общая	3	2,5	7	4,325	27,825
10	3	Алгоритмы сортировки	Опрос	общая	2,5	0,5	2	3,25	31,075
10	3	Алгоритмы сортировки	Практика	простая	3	2	5,5	4,175	35,25
10	4	Алгоритмы поиска	Теория	общая	2,5	6,5	6,5	3,425	38,675
10	4	Алгоритмы поиска	Опрос	общая	2	0,5	1	3,55	42,225
10	4	Алгоритмы поиска	Практика	простая	5,5	5,5	4	3,5	45,725
10	5	Линейные структуры данных: стеки и очередь	Теория	общая	2,5	2,5	7,5	4	49,725
10	5	Линейные структуры данных: стеки и очередь	Опрос	общая	1	0,5	1,5	4,025	53,75
10	5	Линейные структуры данных: стеки и очередь	Практика	средняя	6	1,5	3	4,2	57,95
10	6	Структуры данных и пользовательские типы	Теория	общая	3	4,5	5,5	4	61,95
10	6	Структуры данных и пользовательские типы	Опрос	общая	2	0,5	1	3,775	65,725
10	6	Структуры данных и пользовательские типы	Практика	простая	5	4,5	4	4,075	69,8
10	6	Структуры данных и пользовательские типы	Доп. теория		3,5	4,5	7	3,9	73,7
10	6	Структуры данных и пользовательские типы	Доп. опрос		1,5	0,5	2	3,675	77,375
10	6	Структуры данных и пользовательские типы	Доп. Практика		6,5	5	5,5	3,9	81,275

Источник: составлено автором.

**Приложение Е**  
**(информационное)**

**Вычислительный эксперимент. Вторая группа**

Таблица Е.1 – Вычислительный эксперимент. Вторая группа

Студент	Тема	Тема	Этап	Факт t^CPC в часах	Факт t^Конт преподаватель в часах	Факт t^Конт агент в часах	Факт t^CPPI в часах	Сумма в баллах	Итог в баллах
1	2	3	4	5	6	7	8	9	10
1	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	3	4	1,5	7	3,783	3,783
1	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	2	0,5	0	2	3,55	7,333
1	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	5,5	3,5	1,5	6	3,767	11,1
1	2	Двумерные массивы и алгоритмы обработки матриц	Теория	2	4	1,5	6,5	3,983	15,083
1	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1	0,5	0	1,5	3,8	18,883
1	2	Двумерные массивы и алгоритмы обработки матриц	Практика	6,5	0,5	0,5	2	4,017	22,9
1	3	Алгоритмы сортировки	Теория	2,5	4	2	8,5	3,8	26,7
1	3	Алгоритмы сортировки	Опрос	1,5	0,5	0,5	2,5	3,483	30,183
1	3	Алгоритмы сортировки	Практика	5	2	1	6,5	4,25	34,433
1	4	Алгоритмы поиска	Теория	2	3	1,5	4,5	4,183	38,617
1	4	Алгоритмы поиска	Опрос	1,5	0,5	0	1,5	3,833	42,45
1	4	Алгоритмы поиска	Практика	5,5	3	1	6	3,9	46,35
1	5	Линейные структуры данных: стек и очередь	Теория	3	2	1	8	4,05	50,4
1	5	Линейные структуры данных: стек и очередь	Опрос	1,5	0,5	0	1,5	3,833	54,233
1	5	Линейные структуры данных: стек и очередь	Практика	3,5	4	1,5	4,5	4,067	58,3
1	6	Структуры данных и пользовательские типы	Теория	1,5	4	2	10,5	4,067	62,367
1	6	Структуры данных и пользовательские типы	Опрос	1,5	0,5	0	2	3,983	66,35
1	6	Структуры данных и пользовательские типы	Практика	3,5	4	2	5	4,083	70,433

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
2	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	2,5	3	1	5,5	4,017	4,017
2	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	2	0,5	0	2	3,55	7,567
2	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	4	4,5	2	4,5	3,933	11,5
2	2	Двумерные массивы и алгоритмы обработки матриц	Теория	3	3	1,5	7,5	4,083	15,583
2	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1,5	0,5	0	2	3,783	19,367
2	2	Двумерные массивы и алгоритмы обработки матриц	Практика	5,5	2,5	1	6	3,9	23,267
2	3	Алгоритмы сортировки	Теория	1,5	2	0,5	4,5	4,367	27,633
2	3	Алгоритмы сортировки	Опрос	2,5	0,5	0	2	3,783	31,417
2	3	Алгоритмы сортировки	Практика	5,5	5	2,5	4	3,883	35,3
2	4	Алгоритмы поиска	Теория	4	2,5	1	4	4,117	39,417
2	4	Алгоритмы поиска	Опрос	2	0,5	0	2	3,467	42,883
2	4	Алгоритмы поиска	Практика	6,5	2	1	5	3,883	46,767
2	5	Линейные структуры данных: стек и очередь	Теория	2,5	4	2	10,5	3,717	50,483
2	5	Линейные структуры данных: стек и очередь	Опрос	1	0,5	0	1,5	4,05	54,533
2	5	Линейные структуры данных: стек и очередь	Практика	2,5	3,5	1,5	6	3,8	58,333
2	6	Структуры данных и пользовательские типы	Теория	1	5	2,5	8	3,917	62,25
2	6	Структуры данных и пользовательские типы	Опрос	2	0,5	0	2,5	3,683	65,933
2	6	Структуры данных и пользовательские типы	Практика	4,5	6,5	2,5	6,5	3,683	69,617
3	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	4	0,5	0,5	7	3,9	3,9
3	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0	0	1,5	4,183	8,083
3	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	10	4	1,5	4,5	3,7	11,783
3	2	Двумерные массивы и алгоритмы обработки матриц	Теория	1,5	3	1	7	4,017	15,8

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
3	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	2	0,5	0	1,5	3,85	19,65
3	2	Двумерные массивы и алгоритмы обработки матриц	Практика	4	3	1,5	6	3,8	23,45
3	3	Алгоритмы сортировки	Теория	3	3,5	1,5	10	3,817	27,267
3	3	Алгоритмы сортировки	Опрос	1,5	0,5	0	2,5	3,75	31,017
3	3	Алгоритмы сортировки	Практика	7	0,5	0,5	8	3,817	34,833
3	4	Алгоритмы поиска	Теория	2,5	4	1,5	5,5	4,067	38,9
3	4	Алгоритмы поиска	Опрос	1,5	0,5	0	2	3,783	42,683
3	4	Алгоритмы поиска	Практика	6,5	3	1,5	3,5	3,9	46,583
3	5	Линейные структуры данных: стек и очередь	Теория	3	3	1	5	3,95	50,533
3	5	Линейные структуры данных: стек и очередь	Опрос	2	0,5	0	1	3,9	54,433
3	5	Линейные структуры данных: стек и очередь	Практика	3,5	4	2	2	4,033	58,467
3	6	Структуры данных и пользовательские типы	Теория	4	1,5	0,5	6,5	4,033	62,5
3	6	Структуры данных и пользовательские типы	Опрос	1,5	0,5	0	1,5	3,883	66,383
3	6	Структуры данных и пользовательские типы	Практика	8,5	4	2	5,5	3,75	70,133
4	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	2	3	1,5	9,5	3,9	3,9
4	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0,5	0	1,5	4	7,9
4	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	5,5	4	1,5	7,5	3,633	11,533
4	2	Двумерные массивы и алгоритмы обработки матриц	Теория	1,5	5	2,5	2,5	4,067	15,6
4	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1,5	0,5	0	1	3,967	19,567
4	2	Двумерные массивы и алгоритмы обработки матриц	Практика	5	5	2	2,5	3,917	23,483
4	3	Алгоритмы сортировки	Теория	2	5	2	3	3,95	27,433
4	3	Алгоритмы сортировки	Опрос	1	0,5	0	2	4,133	31,567
4	3	Алгоритмы сортировки	Практика	7,5	3	1	4,5	4,1	35,667
4	4	Алгоритмы поиска	Теория	2,5	4	1,5	8	3,85	39,517
4	4	Алгоритмы поиска	Опрос	1,5	0,5	0	1	3,883	43,4
4	4	Алгоритмы поиска	Практика	8	3	1	4,5	3,817	47,217

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
4	5	Линейные структуры данных: стек и очередь	Теория	2	3	1	6,5	3,933	51,15
4	5	Линейные структуры данных: стек и очередь	Опрос	2	0,5	0	1,5	3,767	54,917
4	5	Линейные структуры данных: стек и очередь	Практика	6	4,5	2	5,5	3,65	58,567
4	6	Структуры данных и пользовательские типы	Теория	2	5	2,5	6	3,9	62,467
4	6	Структуры данных и пользовательские типы	Опрос	3	0,5	0	1,5	3,933	66,4
4	6	Структуры данных и пользовательские типы	Практика	8	3	1,5	5,5	3,9	70,3
5	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	2	2	0,5	6	4,167	4,167
5	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0,5	0	1	3,717	7,883
5	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	7,5	4	1,5	3	3,667	11,55
5	2	Двумерные массивы и алгоритмы обработки матриц	Теория	2,5	1,5	0,5	6	4,183	15,733
5	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1,5	0,5	0	1	3,967	19,7
5	2	Двумерные массивы и алгоритмы обработки матриц	Практика	4	4	1,5	6	3,833	23,533
5	3	Алгоритмы сортировки	Теория	3	3	1	7,5	4,283	27,817
5	3	Алгоритмы сортировки	Опрос	2	0,5	0	1,5	3,733	31,55
5	3	Алгоритмы сортировки	Практика	4	2,5	1	6,5	4,233	35,783
5	4	Алгоритмы поиска	Теория	3	1,5	0,5	7	3,967	39,75
5	4	Алгоритмы поиска	Опрос	1,5	0,5	0	0,5	3,883	43,633
5	4	Алгоритмы поиска	Практика	6,5	2	1	6	4,15	47,783
5	5	Линейные структуры данных: стек и очередь	Теория	2,5	2,5	1	9	3,817	51,6
5	5	Линейные структуры данных: стек и очередь	Опрос	2	0,5	0	2	3,717	55,317
5	5	Линейные структуры данных: стек и очередь	Практика	5	5	2	4,5	3,867	59,183
5	6	Структуры данных и пользовательские типы	Теория	3,5	3,5	1,5	7,5	3,9	63,083
5	6	Структуры данных и пользовательские типы	Опрос	2	0,5	0	2,5	3,85	66,933
5	6	Структуры данных и пользовательские типы	Практика	3,5	5	2,5	5,5	4,133	71,067

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
6	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	3,5	5	2	1,5	3,633	3,633
6	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0,5	0	2	3,95	7,583
6	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	7	4	1,5	6	3,717	11,3
6	2	Двумерные массивы и алгоритмы обработки матриц	Теория	2	3,5	1,5	5	3,883	15,183
6	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	2	0,5	0,5	1,5	3,583	18,767
6	2	Двумерные массивы и алгоритмы обработки матриц	Практика	4	3,5	1,5	5,5	3,65	22,417
6	3	Алгоритмы сортировки	Теория	1,5	4	2	6	3,933	26,35
6	3	Алгоритмы сортировки	Опрос	2	0,5	0	2,5	3,767	30,117
6	3	Алгоритмы сортировки	Практика	8	3,5	1,5	5,5	3,983	34,1
6	4	Алгоритмы поиска	Теория	2,5	2,5	1	10	3,983	38,083
6	4	Алгоритмы поиска	Опрос	1,5	0,5	0	2,5	3,867	41,95
6	4	Алгоритмы поиска	Практика	5	2	1	6	4,2	46,15
6	5	Линейные структуры данных: стек и очередь	Теория	2,5	3	1,5	3,5	3,933	50,083
6	5	Линейные структуры данных: стек и очередь	Опрос	2	0,5	0	1,5	3,683	53,767
6	5	Линейные структуры данных: стек и очередь	Практика	5	4	2	7	3,45	57,217
6	6	Структуры данных и пользовательские типы	Теория	1,5	4	2	8,5	4,117	61,333
6	6	Структуры данных и пользовательские типы	Опрос	1,5	0,5	0	2	3,733	65,067
6	6	Структуры данных и пользовательские типы	Практика	7	3	1	5,5	3,95	69,017
7	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	2,5	3,5	1,5	5,5	3,817	3,817
7	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1	0,5	0	2	3,75	7,567
7	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	4	4	1,5	5,5	3,917	11,483
7	2	Двумерные массивы и алгоритмы обработки матриц	Теория	2,5	4	1,5	3,5	4,133	15,617

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
7	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	0,5	0,5	0	0,5	4,183	19,8
7	2	Двумерные массивы и алгоритмы обработки матриц	Практика	5,5	3,5	1,5	4,5	3,917	23,717
7	3	Алгоритмы сортировки	Теория	2,5	4	1,5	8	3,883	27,6
7	3	Алгоритмы сортировки	Опрос	2	0,5	0	1,5	3,817	31,417
7	3	Алгоритмы сортировки	Практика	4,5	4	2	7,5	3,883	35,3
7	4	Алгоритмы поиска	Теория	2,5	3	1,5	4	4,1	39,4
7	4	Алгоритмы поиска	Опрос	2	0,5	0	2	3,717	43,117
7	4	Алгоритмы поиска	Практика	8	4	2	3	3,917	47,033
7	5	Линейные структуры данных: стек и очередь	Теория	3	3	1	7,5	3,9	50,933
7	5	Линейные структуры данных: стек и очередь	Опрос	2	0,5	0	1,5	3,933	54,867
7	5	Линейные структуры данных: стек и очередь	Практика	5	2	1	4	4,167	59,033
7	6	Структуры данных и пользовательские типы	Теория	4	2,5	1	6	3,783	62,817
7	6	Структуры данных и пользовательские типы	Опрос	1	0,5	0	2	4,133	66,95
7	6	Структуры данных и пользовательские типы	Практика	5,5	4	1,5	5,5	3,967	70,917
8	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	1	2	0,5	3,5	4,45	4,45
8	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0,5	0	2	3,617	8,067
8	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	7	4	2	5	3,533	11,6
8	2	Двумерные массивы и алгоритмы обработки матриц	Теория	2	4	1,5	6	3,817	15,417
8	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1	0,5	0	1,5	3,967	19,383
8	2	Двумерные массивы и алгоритмы обработки матриц	Практика	5,5	3,5	1,5	4,5	3,833	23,217
8	3	Алгоритмы сортировки	Теория	3	2	0,5	11,5	3,883	27,1
8	3	Алгоритмы сортировки	Опрос	2	0,5	0	1,5	3,733	30,833
8	3	Алгоритмы сортировки	Практика	9	3	1,5	1,5	3,783	34,617
8	4	Алгоритмы поиска	Теория	2	2	0,5	3,5	4,3	38,917
8	4	Алгоритмы поиска	Опрос	1,5	0,5	0	2	3,7	42,617
8	4	Алгоритмы поиска	Практика	3,5	3,5	1,5	3,5	4,033	46,65

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
8	5	Линейные структуры данных: стек и очередь	Теория	4	3	1,5	7,5	3,85	50,5
8	5	Линейные структуры данных: стек и очередь	Опрос	1,5	0,5	0	1	4,05	54,55
8	5	Линейные структуры данных: стек и очередь	Практика	4,5	5	2,5	4,5	3,783	58,333
8	6	Структуры данных и пользовательские типы	Теория	3,5	4	2	11	3,7	62,033
8	6	Структуры данных и пользовательские типы	Опрос	2	0,5	0,5	2	3,567	65,6
8	6	Структуры данных и пользовательские типы	Практика	7	3,5	1,5	2,5	4,267	69,867
9	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	2	3,5	1,5	3	4,2	4,2
9	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0,5	0	1,5	3,75	7,95
9	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	3,5	1,5	0,5	7	3,95	11,9
9	2	Двумерные массивы и алгоритмы обработки матриц	Теория	3	3,5	1,5	5	4,083	15,983
9	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1,5	0,5	0	1,5	4	19,983
9	2	Двумерные массивы и алгоритмы обработки матриц	Практика	8,5	3,5	1,5	4,5	4,017	24
9	3	Алгоритмы сортировки	Теория	2	5	2	5	4,117	28,117
9	3	Алгоритмы сортировки	Опрос	1	0,5	0	1	4,017	32,133
9	3	Алгоритмы сортировки	Практика	6	5	2	6	3,75	35,883
9	4	Алгоритмы поиска	Теория	3,5	3	1,5	5	3,9	39,783
9	4	Алгоритмы поиска	Опрос	1,5	0,5	0	2,5	3,533	43,317
9	4	Алгоритмы поиска	Практика	4,5	4	2	4,5	3,917	47,233
9	5	Линейные структуры данных: стек и очередь	Теория	2	3	1	8	3,783	51,017
9	5	Линейные структуры данных: стек и очередь	Опрос	1,5	0,5	0	1,5	3,75	54,767
9	5	Линейные структуры данных: стек и очередь	Практика	8,5	4	1,5	4	3,85	58,617
9	6	Структуры данных и пользовательские типы	Теория	3	4,5	2	8	3,75	62,367
9	6	Структуры данных и пользовательские типы	Опрос	1	0,5	0	1,5	4,033	66,4
9	6	Структуры данных и пользовательские типы	Практика	3,5	4	1,5	7,5	4,117	70,517

Продолжение таблицы Е.1

1	2	3	4	5	6	7	8	9	10
10	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Теория	1,5	2,5	1	5	4,233	4,233
10	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Опрос	1,5	0,5	0	2	3,617	7,85
10	1	Линейные структуры данных и алгоритмы обработки массивов и строк	Практика	4,5	3	1	4	4,267	12,117
10	2	Двумерные массивы и алгоритмы обработки матриц	Теория	2	2	1	7	4,35	16,467
10	2	Двумерные массивы и алгоритмы обработки матриц	Опрос	1,5	0,5	0	1	4,05	20,517
10	2	Двумерные массивы и алгоритмы обработки матриц	Практика	6	3,5	1,5	2	3,8	24,317
10	3	Алгоритмы сортировки	Теория	4	5	2	9,5	3,317	27,633
10	3	Алгоритмы сортировки	Опрос	1	0	0	2	4,067	31,7
10	3	Алгоритмы сортировки	Практика	6	6,5	2,5	6,5	3,867	35,567
10	4	Алгоритмы поиска	Теория	2,5	2,5	1	5	4,183	39,75
10	4	Алгоритмы поиска	Опрос	2	0,5	0	2	3,717	43,467
10	4	Алгоритмы поиска	Практика	6	3,5	1,5	3	4,05	47,517
10	5	Линейные структуры данных: стек и очередь	Теория	2	3	1	7,5	4,2	51,717
10	5	Линейные структуры данных: стек и очередь	Опрос	1,5	0,5	0	1	3,967	55,683
10	5	Линейные структуры данных: стек и очередь	Практика	3,5	3,5	1,5	6	3,817	59,5
10	6	Структуры данных и пользовательские типы	Теория	2,5	3,5	1,5	5,5	4,017	63,517
10	6	Структуры данных и пользовательские типы	Опрос	2,5	0,5	0	2	3,617	67,133
10	6	Структуры данных и пользовательские типы	Практика	3,5	4	2	4,5	4,083	71,217

Источник: составлено автором.

**Приложение Ж**  
**(информационное)**  
**Результаты опроса студентов и преподавателей**

Таблица Ж.1 – Результаты опроса среди студентов и преподавателей на тему «Оценка готовности к применению нейросетей глубокого обучения в электронном обучении»,

Студенты	1 Владеете ли вы знаниями языков программирования и ИТ-инструментов?	2 Если «нет», планируете в будущем получить знания по программированию?	3 Знакомы ли вы с НГО, БЯМ и концепцией «НГО как оценщик»	4 Если «нет», хотели бы вы изучать основы программирования с применением НГО, БЯМ и концепцией «НГО как оценщик»	5 Если «да», то есть ли у вас желание получать более глубокие знания с применением НГО, БЯМ и концепцией «НГО как оценщик»	6 Считаете ли что ваши знания улучшатся, если НГО будет разбирать (объяснять, проверять результаты) задачи из учебного курса?
1	2	3	4	5	6	7
БИИК СибГУТИ						
да	12	20	7	12	8	27
нет	25	10	30	9	13	6
затрудняюсь ответить	-	7	-	16	16	4
всего	37	37	37	37	37	37
ВСГУТУ						
да	34	24	8	17	11	23
нет	16	12	42	19	7	3
затрудняюсь ответить	-	14	-	14	16	13
всего	50	50	50	50	50	50

Продолжение таблицы Ж.1

1	2	3	4	5	6	7
БГСХА						
да	15	17	16	15	12	14
нет	18	7	17	7	8	3
затрудняюсь ответить	-	9	-	11	5	8
всего	33	33	33	33	33	33
Колледж ВСГУТУ						
да	40	8	3	20	2	42
нет	14	2	51	6	1	2
затрудняюсь ответить	-	4	-	25	51	10
всего	54	54	54	54	54	54
Итого						
всего	174	174	174	174	174	174
да процент	58,0	51,5	19,5	37,4	22,0	68,4
нет процент	42,0	23,1	80,5	24,0	19,3	7,7
затрудняюсь ответить процент	0	19,5	0	37,9	51,5	16,7

Источник: составлено автором.

Таблица Ж.2 – Результаты опроса преподавателей

Преподаватели	1 Владеете ли вы знаниями языков программирования и ИТ-инструментов?	2 Если «нет», планируете в будущем получить знания программирования?	3 Знакомы ли вы с НГО, БЯМ и концепцией «НГО как оценщик»	4 Если «нет», хотели бы вы изучать основы программирования с применением НГО, БЯМ и концепцией «НГО как оценщик»	5 Если «да», то есть ли у вас желание получать более глубокие знания с применением НГО, БЯМ и концепцией «НГО как оценщик»	6 Считаете ли что ваши знания улучшатся, если НГО будет разбирать (объяснять, проверять результаты) задачи из учебного курса?
1	2	3	4	5	6	7
БИИК СибГУТИ						
да	3	1	-	1	4	4
нет	1		-	1		
затрудняюсь ответить	-	3	4	2	-	-
всего	4	4	4	4	4	4
ВСГУТУ						
да	3	4	3	2	7	7
нет	6	3	2	3	-	-
затрудняюсь ответить		2	4	4	2	2
всего	9	9	9	9	9	9
БГСХА						
да	7	5	4	2	5	3
нет	4		1	3		
затрудняюсь ответить	-	3	4	6	6	4

Продолжение таблицы Ж.2

1	2	3	4	5	6	7
всего	11	8	9	11	11	7
Колледж ВСГУТУ						
да	5	1	2	3	1	2
нет	2	1	5	2	1	3
затрудняюсь ответить	-	-	-	-	-	2
всего	7	2	7	5	2	7
Итого						
всего	31	31	31	31	31	31
да процент	58,1	47,8	33,3	30,8	70,8	81,0
нет процент	41,9	17,4	22,2	23,1	4,2	0,0
затрудняюсь ответить процент	0	34,8	44,4	46,2	25,0	19,0

Источник: составлено автором.

## Приложение И

(Информационное)

### Формирование контекста агента на Python

```

import json
import requests
from sentence_transformers import SentenceTransformer
from llama_cpp import Llama
from bs4 import BeautifulSoup
from keyword_search import KeywordSearch

# Загрузка конфигурации
with open('config.json', 'r', encoding='utf-8') as config_file:
    config = json.load(config_file)

# Загрузка модели Sentence-BERT для векторизации
sentence_model = SentenceTransformer(config['sentence_model_name'])

# Попытка загрузки модели с GPU, с откатом на CPU в случае ошибки
try:
    print("Пытаемся загрузить модель с использованием GPU (CUDA).")
    gguf_model = Llama(
        model_path=config['gguf_model_path'],
        n_ctx=config['gguf_model_params']['n_ctx'],
        n_threads=config['gguf_model_params']['n_threads'],
        n_gpu_layers=32
    )
    print("Модель успешно загружена с использованием GPU.")
except Exception as e:
    print(f"Не удалось загрузить модель с GPU: {e}")
    print("Переключаемся на CPU.")
    gguf_model = Llama(
        model_path=config['gguf_model_path'],
        n_ctx=config['gguf_model_params']['n_ctx'],
        n_threads=config['gguf_model_params']['n_threads']
    )

# Настройки OpenSearch
BASE_URL = config['opensearch']['base_url']
API_ENDPOINT = config['opensearch']['endpoint']
HEADERS = {'Content-Type': 'application/json'}

# Максимальная длина контекста для Llama
MAX_CONTEXT_LENGTH = 8000 # Оставляем запас

keyword_search = KeywordSearch()

class SearchElement:
    def __init__(self, question, source, top_questions):
        self.question = question
        self.source = source
        self.top_questions = top_questions

    def truncate_text(text, max_length):
        return text[:max_length] # Берем только первые N символов

    def clean_html(text):
        """Удаляет HTML-теги из текста."""
        return BeautifulSoup(text, "html.parser").get_text()

```

```

def search_elem_to_prompt(search_elem):
    return f"Вопрос: {search_elem.question} \nОтвет:
{search_elem.source}\n"

def search_in_opensearch(question: str, top_k=5):
    # Используем шаблон payload из конфига и подставляем question напрямую
    payload_template = config['opensearch']['payload']
    payload = payload_template.copy()
    payload["searchText"] = question

    url = f"{BASE_URL}{API_ENDPOINT}"
    params = config['opensearch']['params']
    params["perPage"] = top_k

    response = requests.post(url, json=payload, headers=HEADERS,
params=params)
    if response.status_code != 200:
        raise Exception("Ошибка запроса к OpenSearch")

    data = response.json()
    hits = data.get("results", [])[0].get("hits", {}).get("hits", [])

    search_results = [
        SearchElement(
            question=hit['_source']['name'],
            source=clean_html(hit['_source'].get("content", "Ответ не
найден")),
            top_questions=hit['_source']['name']
        ) for hit in hits[:top_k]
    ]

    if not search_results:
        return keyword_search.keyword_main(question)

    return search_results

# Главная функция вопрос-ответной системы
def question_answer_system(user_input):
    search_results = search_in_opensearch(user_input)

    prompt = ".join(search_elem_to_prompt(elem) for elem in
search_results)
    prompt = truncate_text(prompt, MAX_CONTEXT_LENGTH)

    try:
        final_answer = gguf_model(
            config['prompt_template'].format(raw_answer=prompt,
question=user_input),
            max_tokens=config['generation_params']['max_tokens'],
            temperature=config['generation_params']['temperature'],
repeat_penalty=config['generation_params']['repeat_penalty'],
            )["choices"][0]["text"].strip()

        # Очистка ответа от лишних повторов вопроса или "Ответ:"
        final_answer = final_answer.replace(f"Вопрос: {user_input}",
"").strip()
        final_answer = final_answer.replace("Ответ:", "").strip()
        # Удаление лишних переносов строк и пробелов
        final_answer = " ".join(final_answer.split())
    except Exception as e:

```

```
print(f"Ошибка при генерации ответа: {e}")
final_answer = "Ошибка генерации ответа"

# Сохранение в JSON вместо output.txt
output_data = {
    "question": user_input,
    "answer": final_answer,
    "top_questions": [elem.question for elem in search_results]
}

with open('output.json', 'w', encoding='utf-8') as f:
    json.dump(output_data, f, ensure_ascii=False, indent=4)

print("Данные сохранены в файл 'output.json'.")

return final_answer, search_results

# Пример использования
if __name__ == "__main__":
    user_question = input("Введите ваш вопрос: ")
    try:
        response, search_results = question_answer_system(user_question)
        print(response)
        gguf_model.close()
    except Exception as e:
        print(f"Произошла ошибка: {str(e)}")
```