

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: Ca+D95LgZGfyuDQQ7XRJa2KBZEz01VSp
Дата: 25.11.2025 г.

Е.П. Догадина

Проектный практикум

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике
и финансах»

Профиль:

«Прикладные информационные системы в экономике и финансах»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2026

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	8
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	8
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	9
5.1.	Содержание дисциплины	9
5.2.	Учебно-тематический план	11
5.3.	Содержание семинаров	12
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	15
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	15
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	18
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	23
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	34
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	35
10.	Методические указания для обучающихся по освоению дисциплины	36
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	36
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	38

1. Наименование дисциплины

«Проектный практикум».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПKN-2	Способность разрабатывать алгоритмы и программы с использованием современных технологий программирования	Владеет объектно-ориентированным языком программирования на уровне знания синтаксиса и семантики, основ стандартной библиотеки	знать: принципы инкапсуляции, наследования и полиморфизма; базовой синтаксис выбранного языка программирования; принципы проектирования и методологии разработки уметь: применять принципы объектно-ориентированного программирования в практике; работать над групповыми проектами, включая планирование, распределение задач и совместную разработку кода; создавать тесты для проверки функциональности программ
		Использует инструментальные средства программирования (IDE, SDK, API, популярные фреймворки и библиотеки)	знать: популярные SDK (Software Development Kit) и их роли в разработке приложений для конкретных платформ; основные понятия API, типы API (REST, SOAP и т.д.) и их применение в разработке; популярные фреймворки, связанные с выбранным языком программирования (Django, Flask для Python, Spring для Java, React, Angular для JavaScript); этапы разработки программного обеспечения, включая проектирование, реализацию, тестирование и деплой уметь: применять инструменты SDK для создания, тестирования и отладки приложений; организовывать запросы к API для

			получения и отправки данных, обработки ответов и интеграции функциональности в приложения; использовать системы контроля версий для отслеживания изменений в коде и управления совместной работой
		Организовывает кодовую базу, ориентируется в существующем коде, демонстрирует знание общепринятых соглашений и политик в области оформления кода	знать: стандарты оформления кода; инструменты для статического анализа кода уметь: создавать и поддерживать структуру проекта, соответствующую требованиям и стандартам; использовать линтеры и формatters для автоматической проверки и исправления оформления кода
		Проектирует текстовый, программный или графический интерфейс программной системы исходя из ее назначения	знать: ключевых элементов интерфейса; методологии проектирования интерфейсов (Design Thinking, Agile UX, User-Centered Design); принципы доступности (Accessibility) и инклюзивного дизайна, обеспечивающие доступность интерфейсов для всех пользователей уметь: использовать принципы визуального дизайна; разрабатывать графические элементы интерфейса; проектировать текстовые интерфейсы, обеспечивая их удобство и читабельность; создавать программные интерфейсы (API), учитывая требования к взаимодействию между различными компонентами системы
ПКН-3	Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов	знать: типы структур данных и их применение; основные принципы разработки алгоритмов уметь: использовать простые структуры данных для хранения и обработки информации в программных проектах; адаптировать и модифицировать алгоритмы для решения специфических задач; оптимизировать существующий код,

			используя знания о сложностях алгоритмов и структур данных
		Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе	знать: популярные стандарты и методологии сбора требований; принципы работы с требованиями в рамках методологий Agile и Scrum; инструменты и программное обеспечение для документирования и управления требованиями уметь: классифицировать и приоритизировать требования в зависимости от их важности и влияния на проект; оценивать архитектурные паттерны и выбирать подходящие решения, основываясь на собранных требованиях; проводить проверки и оценку системы на соответствие функциональным и нефункциональным требованиям в процессе разработки
		Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии	знать: основы одного или нескольких объектно-ориентированных языков программирования; особенности платформы или технологии, на которой будет разрабатываться система; методы и подходы к декомпозиции задач; архитектурные паттерны уметь: создавать интерфейсы и абстрактные классы для определения контрактов между компонентами системы; проводить юнит-тестирование и интеграционное тестирование; создавать диаграммы для визуализации структуры системы и взаимосвязей между классами
ПKN-5	Способность участвовать в документальном сопровождении разработки ИТ в рамках проектных групп, применять	Демонстрирует знание основ версионирования и управления изменениями при разработке ПО. Использует системы контроля версий для ведения совместной разработки	знать: принципы работы с веткам в системах контроля версий и стратегии их использования в командной разработке; принципы работы распределенных и централизованных систем контроля версий уметь: создавать и настраивать репозитории в системах контроля

	средства автоматизации управления проектами ИТ		версий; проводить ревью кода; использовать инструменты для автоматизации управления версиями и интеграции для обеспечения CI/CD (непрерывной интеграции и доставки)
	Демонстрирует знание основ тестирования программного обеспечения, умение создавать автоматизированные модульные и интеграционные тесты		знать: виды тестирования; принципы модульного тестирования; популярные инструменты и фреймворки для автоматизированного тестирования уметь: разрабатывать интеграционные тесты для проверки взаимодействия между несколькими компонентами системы; настраивать тестовые окружения; принципы интегрирования тестов в процесс непрерывной интеграции и доставки (CI/CD); использовать инструменты для анализа и диагностики проблем
	Готовит документацию к программе, коммуницирует в пределах группы разработки и за ее границами о значимых аспектах информационной системы и информационной инфраструктуры в письменной и устной форме		знать: принципы работы в команде, включая роли и ответственности участников, а также методы ведения обсуждений и собраний; методы сбора и анализа отзывов от пользователей и членов команды; принципы визуализации данных и использования диаграмм для пояснения архитектуры системы уметь: оформлять документацию в соответствии с установленными стандартами и требованиями; внедрять изменения и обновления в документацию на основе полученной информации
	Демонстрирует знание жизненного цикла информационных систем, участвует в процессе разработки ПО на разных этапах		знать: методы мониторинга и контроля рисков на протяжении всего жизненного цикла проекта; основные критерии и метрики, по которым оценивается успешность разработки и внедрения информационной системы уметь: проектировать архитектуру системы и разрабатывать функциональные компоненты; анализировать требования пользователей и бизнеса, а также

			формализовать их в виде документации
--	--	--	--------------------------------------

3. Место дисциплины в структуре образовательной программы

Дисциплина «Проектный практикум» относится к «Общефакультетскому (предпрофильному) циклу».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 3 (в часах)	Семестр 4 (в часах)	Семестр 5 (в часах)	Семестр 6 (в часах)	Семестр 7 (в часах)
Общая трудоёмкость дисциплины	5/180	36	36	36	36	36
Контактная работа- Аудиторные занятия	80	16	16	16	16	16
Лекции	0					
Семинары, практические занятия	80	16	16	16	16	16
Самостоятельная работа	100	20	20	20	20	20
Вид текущего контроля	Проектная работа, Проектная работа, Проектная работа, Проектная работа, Проектная работа	Проектная работа	Проектная работа	Проектная работа	Проектная работа	Проектная работа
Вид промежуточной аттестации	Зачет, Зачет, Зачет, Зачет, Зачет	Зачет	Зачет	Зачет	Зачет	Зачет

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в дисциплину "Проектный практикум"

Содержание образовательной программы, социальные партнеры, научные школы.. Ключевые компетенции, формируемые при обучении по направлению подготовки «Прикладная информатика». Направления трудоустройства и карьерные траектории выпускников направления «Прикладная информатика».

Тема 2. История развития прикладной информатики и текущий технологический ландшафт отрасли

Основные вехи развития прикладной информатики. Знаковые кейсы создания и использования передовых моделей прикладной информатики. Обзор текущего ландшафта прикладной информатики и ближайших перспектив ее развития. Основные вехи в развитии: теоретических основ ИТ, аппаратных решений в области ИТ, программных архитектур, методологий разработки ПО и концепция программирования. Обзор текущего состояния ИТ и современных тенденций в области ИТ.

Тема 3. Основы проектного менеджмента

Введение в проектный менеджмент: основные понятия и процессы. Методы оценки и управления рисками в проектах. Использование Agile и Scrum в управлении проектами.

Тема 4. Жизненный цикл разработки программного обеспечения

Этапы жизненного цикла разработки ПО: от идеи до внедрения. Модели разработки ПО: каскадная, Agile, V-модель и другие. Документирование процессов разработки: спецификации, требования, тестирование.

Тема 5. Анализ и проектирование информационных систем

Сбор и анализ требований: методы и техники. Моделирование бизнес-процессов с использованием BPMN. Проектирование архитектуры информационной системы: UML-диаграммы и другие инструменты.

Тема 6. Прототипирование и создание интерфейсов

Основы UX/UI-дизайна: принципы и лучшие практики. Инструменты для прототипирования интерфейсов: Figma, Axure, Balsamiq. Проведение юзабилити-тестирования прототипов.

Тема 7. Современные технологии в разработке

Введение в облачные технологии и их применение в проектах. Основы DevOps: интеграция разработки и операций. Использование контейнеризации (Docker) в разработке.

Тема 8. Этические и правовые аспекты в разработке ПО

Вопросы авторского права и лицензирования программного обеспечения. Этические аспекты разработки и использования информационных технологий. Защита данных и безопасность информации в проектах.

Тема 9. Командная работа и коммуникация

Эффективная команда: роли и ответственности участников. Коммуникационные навыки: как вести переговоры и обсуждения. Управление конфликтами и построение командной культуры.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Введение в дисциплину "Проектный практикум"	14	4	0	4	10
2	История развития прикладной информатики и текущий технологический ландшафт отрасли	16	6	0	6	10
3	Основы проектного менеджмента	20	10	0	10	10
4	Жизненный цикл разработки программного обеспечения	20	10	0	10	10
5	Анализ и проектирование информационных систем	30	10	0	10	20
6	Прототипирование и создание интерфейсов	20	10	0	10	10
7	Современные технологии в разработке	20	10	0	10	10
8	Этические и правовые аспекты в разработке ПО	20	10	0	10	10
9	Командная работа и коммуникация	20	10	0	10	10
	Итого	180	80	0	80	100

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в дисциплину "Проектный практикум"	- Что такое проект и как он отличается от обычной деятельности?- Какие основные характеристики проекта определяют его успешность?- Какие этапы включает в себя жизненный цикл проекта?- Какие роли могут существовать в команде проекта, и каковы их основные обязанности?- Как важно распределение ролей для успешной реализации проекта?- Как взаимодействуют разные роли в команде?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
История развития прикладной информатики и текущий технологический ландшафт отрасли	- Какие известные компании или стартапы в области прикладной информатики вы считаете успешными и почему?- Каковы примеры революционных проектов или технологий, которые изменили индустрию?- Как опыт успешных и неудачных проектов может помочь студентам в их будущей карьере?- Как прикладная информатика влияет на бизнес-процессы и принятие решений в современных организациях?- Как технологии меняют общественные нормы и поведение пользователей?- Какие основные угрозы безопасности существуют в современном цифровом мире и как с ними справляться?- Каковы этические вопросы, связанные с использованием технологий в прикладной информатике?- Как конфиденциальность данных становится важным аспектом разработки программного обеспечения?- Какие технологии, по вашему мнению, будут определять будущее отрасли?- Как студенты могут подготовиться к будущим изменениям и вызовам в сфере прикладной информатики?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Основы проектного менеджмента	Что отличает проект от операционной деятельности?Приведите примеры успешных и неудачных проектов. В чём причины их исхода? Какие признаки характеризуют проект как уникальный, временный и целенаправленный? Почему проектный подход становится всё более востребованным в современных организациях?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Жизненный цикл разработки	Какие фазы выделяются в жизненном цикле проекта? Чем они отличаются?Какие документы	Индивидуальное выполнение заданий,

программного обеспечения	разрабатываются на каждой фазе проекта? В чём разница между инициацией и планированием проекта? Почему важно формально завершать проект, даже если он признан неудачным?	групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Анализ и проектирование информационных систем	Зачем моделировать «как есть» (As-Is) и «как должно быть» (To-Be)? Какие нотации используются для моделирования (BPMN, IDEF0, UML, DFD)? В чём их сильные и слабые стороны? Какие ошибки чаще всего допускают при описании бизнес-процессов? Можно ли автоматизировать процесс, который плохо организован в реальности?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Прототипирование и создание интерфейсов	Почему UX/UI-дизайн — неотъемлемая часть проектирования ИС? Какие принципы лежат в основе удобного интерфейса (юзабилити, доступность, согласованность)? Какие инструменты используются для прототипирования (Figma, Adobe XD, Balsamiq)? Как учитывать разнообразие пользователей (роли, уровень подготовки, устройства)? Для чего нужен прототип? Чем он отличается от макета и MVP? В чём разница между низкоуровневым (low-fidelity) и высокоуровневым (high-fidelity) прототипами? Когда стоит использовать бумажный прототип, а когда — интерактивный цифровой? Как прототипирование помогает избежать дорогостоящих ошибок на этапе разработки?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Современные технологии в разработке	Какие методологии проектирования вы знаете (RAD, RUP, Agile/Scrum, DevOps)? Как стандарты типа ISO/IEC 25010 помогают оценивать качество ИС? В чём разница между проектированием «с нуля» и реинжинирингом существующей системы? Как документировать архитектуру и требования (например, с помощью шаблонов IEEE)?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Этические и правовые аспекты в разработке ПО	Что значит быть «этичным разработчиком»? Какие принципы лежат в основе профессиональной этики (например, Кодекс ACM/IEEE)? Может ли программист нести моральную ответственность за последствия использования его кода? Как поступать, если заказчик требует реализовать функционал, нарушающий права пользователей или	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные

	закон Должен ли разработчик отказываться от участия в проектах, связанных с военными технологиями, слежкой или манипуляцией?	технологии).
Командная работа и коммуникация	Как формируется эффективная проектная команда? Какие барьеры могут возникать в коммуникации внутри проекта? Какие инструменты используются для планирования коммуникаций? Как мотивировать команду в условиях неопределённости и давления сроков? Должны ли курсы по этике и праву быть обязательными в ИТ-образовании? Как компании могут внедрять этические принципы в корпоративную культуру (этические комитеты, кодексы поведения)? Должен ли разработчик проходить обучение по защите данных и ИБ?	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в дисциплину "Проектный практикум"	Как проектный практикум связан с будущей профессиональной деятельностью? Что такое «проект» в контексте образовательного процесса? Какие типы проектов существуют (учебные, исследовательские, социальные, технические и др.)? Что включает в себя проектная документация? Какие методы и инструменты используются при управлении проектами?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
История развития прикладной информатики и текущий технологический ландшафт отрасли	Какие основные направления существуют в современной прикладной информатике? Какие технологии сегодня считаются ключевыми в прикладной информатике (облачные вычисления, большие данные, ИИ, IoT и др.)? Что такое «цифровая трансформация» и как прикладная информатика способствует её реализации? Как изменились требования к специалистам в области прикладной информатики за последние 10–15 лет? Какие современные платформы, фреймворки и инструменты наиболее востребованы в индустрии? Как глобальные тренды (удалённая работа, устойчивое развитие, кибербезопасность) влияют на развитие отрасли? Какова роль open-source сообществ и экосистем в современной прикладной информатике? Какие отрасли активнее всего применяют решения в области прикладной информатики (финансы, здравоохранение, логистика, образование и др.)?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Основы проектного менеджмента	Что включает управление содержанием (scope management) проекта? Что такое «раздутие требований» (scope creep)? Как планируется и контролируется расписание (time management) проекта? Как осуществляется управление стоимостью (cost management) проекта? Какие методы используются для управления рисками проекта?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Жизненный цикл	Какие задачи решаются на этапе сбора и анализа	Индивидуальное

разработки программного обеспечения	требований? Кто в этом участвует?Что включает в себя проектирование архитектуры и интерфейсов ПО?Какие виды тестирования применяются на этапе валидации и верификации?Что происходит на этапе внедрения (развёртывания) программного продукта?Почему этап сопровождения и поддержки может быть самым длительным и затратным?В чём особенности каскадной (водопадной) модели SDLC? Каковы её преимущества и недостатки?Что такое итеративная модель разработки? Чем она отличается от каскадной?Как работает спиральная модель? В каких случаях она наиболее эффективна?Что представляет собой V-модель разработки? Как она связывает этапы разработки и тестирования?Какие недостатки классических моделей привели к появлению гибких (Agile) подходов?	выполнение заданий с использованием соответствующего программного обеспечения.
Анализ и проектирование информационных систем	Кто такие заинтересованные стороны (стейкхолдеры) ИС и как с ними взаимодействовать?Какие виды требований выделяют (функциональные, нефункциональные, бизнес-требования)?Какие методы используются для сбора требований (интервью, анкетирование, наблюдение, анализ документов)?Что такое use case (прецедент) и как он помогает в анализе требований?Как документируются требования к ИС (SRS — Software Requirements Specification)? Какие стандарты существуют?В чём разница между логическим и физическим проектированием ИС?Какие виды архитектур ИС существуют (одноуровневая, клиент-сервер, трёхзвенная, микросервисная и др.)?Какую роль играют диаграммы в проектировании ИС? Какие нотации используются (UML, BPMN, ERD)?Что такое модель «сущность–связь» (ER-модель) и как она применяется при проектировании баз данных?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Прототипирование и создание интерфейсов	Что такое пользовательский интерфейс (UI) и чем он отличается от пользовательского опыта (UX)? Какова роль прототипирования в процессе разработки программного обеспечения или цифрового продукта?Почему важно создавать прототипы до начала программирования?Какие цели преследует этап проектирования интерфейсов?Как плохой интерфейс может повлиять на успех всего проекта?В чём разница между низкодетализированным (low-fi) и высокодетализированным (high-fi) прототипом? Что такое wireframe и когда его используют?Какие	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.

	существуют подходы к прототипированию: горизонтальное vs вертикальное?Что такое интерактивный прототип и почему он важен для тестирования?Какие инструменты используются для создания прототипов (Figma, Adobe XD, Axure, Balsamiq, Sketch и др.)?	
Современные технологии в разработке	В чём разница между классическими (водопадными) и гибкими (Agile, Scrum, Kanban) методологиями управления проектами?Когда целесообразно применять Agile, а когда — традиционный подход?Какие инструменты планирования и контроля проектов вы знаете (Gantt-диаграммы, диаграммыPERT/CPM, burndown-чарты и др.)?Какие цифровые инструменты (Trello, Jira, Microsoft Project, Asana и т. п.) используются для управления проектами сегодня?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Этические и правовые аспекты в разработке ПО	В чём разница между этическими нормами и юридическими требованиями в IT-сфере?Какие профессиональные кодексы этики существуют для IT-специалистов (например, ACM, IEEE)?Может ли программный продукт быть технически корректным, но этически неприемлемым? Приведите примеры. Как глобализация и цифровизация усиливают ответственность разработчиков ПО?Какие этические дилеммы могут возникнуть при разработке систем ИИ (например, предвзятость алгоритмов)?Как обеспечить справедливость и недискриминацию в алгоритмах и интерфейсах?Что такое «технологическая ответственность» и как она проявляется в повседневной работе разработчика? Должен ли разработчик отказываться от участия в создании ПО, которое может нанести вред обществу?Как учитывать этические последствия при проектировании систем слежения, распознавания лиц или автоматизированного принятия решений?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Командная работа и коммуникация	Какие роли существуют в проектной команде? Что делает проектный менеджер?Кто такие «стейкхолдеры» (заинтересованные стороны) и как с ними взаимодействовать?Почему эффективные коммуникации критически важны для успеха проекта?Какие навыки (hard и soft skills) должен иметь успешный проектный менеджер?Как разрешать конфликты в проектной команде?	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания к проектной работе 3 семестра

1. Формирование проектной команды и распределение ролей. Составьте оргструктуру команды (PM, аналитик, разработчики, тестировщики и др.), опишите обязанности и компетенции каждого.
2. Постановка SMART-целей проекта. Выберите тему (например, «Организация хакатона в вузе») и сформулируйте цели по критериям SMART.
3. Анализ рисков учебного проекта. Создайте реестр рисков: идентифицируйте, оцените по вероятности и влиянию, предложите стратегии реагирования (избежание, передача, снижение, принятие).
4. Разработка WBS (Work Breakdown Structure). Для проекта «Создание лендинга для образовательного курса» разбейте работу на иерархические пакеты работ до уровня задач.
5. Построение диаграммы Ганта вручную или с помощью инструмента. Спланируйте сроки выполнения задач проекта «Организация онлайн-конференции», определите зависимости и критический путь.
6. Оценка трудозатрат по методу PERT или экспертной оценке. Оцените длительность ключевых задач проекта, используя оптимистичную, пессимистичную и наиболее вероятную оценки.
7. Составление бюджета проекта с учётом резервов. Рассчитайте прямые и косвенные расходы, включите буфер на непредвиденные расходы (например, для проекта «Разработка чат-бота»).

Примерные задания к проектной работе 4 семестра:

1. Сравнение применения Waterfall и Agile в одном сценарии. Возьмите проект (например, «Разработка справочника университета») и опишите, как он реализовывался бы в каждой модели - с плюсами и минусами.
2. Планирование итерации (спринта) по методологии Scrum. Составьте бэклог, определите цели спринта, оцените story points, проведите планирование.

3. Создание отчёта о завершении проекта (Project Closure Report). Включите: достигнутые цели, отклонения, уроки, благодарности, рекомендации для будущих проектов.
4. Проведение ретроспективы учебного проекта. Организуйте сессию «Что пошло хорошо / Что можно улучшить», зафиксируйте выводы и действия.
5. Разработка чек-листа «Готов ли проект к запуску?». Создайте инструмент для финальной проверки: документация, тестирование, обучение пользователей, резервное копирование, утверждение стейкхолдерами и т.д.

Примерные задания к проектной работе 5 семестра:

1. Создание wireframe для мобильного приложения "Ежедневник задач". Разработайте низкодетализированный прототип основных экранов (список задач, добавление задачи, профиль пользователя).
2. Редизайн главной страницы существующего сайта. Выберите любой публичный сайт (например, новостной портал) и предложите улучшенную версию главной страницы с акцентом на читаемость и навигацию.
3. Прототип веб-формы регистрации. Создайте интерактивный прототип формы регистрации с валидацией полей, подсказками и обработкой ошибок.
4. Интерфейс для поиска книг в библиотеке. Спроектируйте простой интерфейс для поиска и отображения книг по автору, названию и жанру (веб или мобильная версия).
5. Создание дизайн-системы для мини-приложения. Определите цветовую палитру, типографику, кнопки и компоненты для вымышленного сервиса (например, доставка еды).
6. Интерфейс для образовательной платформы (LMS). Создайте прототип личного кабинета студента с курсами, прогрессом, уведомлениями и календарём занятий.
7. Прототип каршерингового приложения. Включите: карту с автомобилями, бронирование, разблокировку, оплату и завершение поездки.
8. Портал для поиска волонёрских мероприятий. Спроектируйте фильтрацию по локации, типу активности, дате, а также страницу мероприятия с описанием и кнопкой «Принять участие».

9. Дизайн интерфейса для людей с нарушениями зрения. Разработайте версию приложения с учётом принципов доступности: контраст, навигация с клавиатуры, поддержка скринридеров.
10. Прототип голосового помощника для управления умным домом. Спроектируйте визуальный интерфейс, отражающий взаимодействие с голосовым ИИ (история запросов, подтверждения, ошибки).
11. Создание дизайн-системы и компонентной библиотеки для стартапа. Разработайте полноценную дизайн-систему: токены, компоненты, состояния, гайд по использованию.

Примерные задания к проектной работе 6 семестра:

1. Создание веб-приложения с использованием React. Разработайте простое веб-приложение на React, реализующее функциональность ToDo-листа. Изучите компоненты, состояние и маршрутизацию.
2. Разработка RESTful API с использованием Node.js. Создайте RESTful API для управления данными о пользователях. Используйте Express.js и MongoDB для хранения информации.
3. Мобильное приложение на Flutter. Разработайте кросс-платформенное мобильное приложение на Flutter, которое отображает список книг с возможностью добавления новых записей.
4. Внедрение CI/CD с помощью GitHub Actions. Настройте процесс непрерывной интеграции и доставки (CI/CD) для вашего проекта на GitHub с использованием GitHub Actions.
5. Реализация чат-приложения на WebSocket. Создайте простое чат-приложение с использованием WebSocket для обмена сообщениями в реальном времени.
6. Применение машинного обучения в проекте. Разработайте приложение, использующее алгоритмы машинного обучения для предсказания цен на жилье. Используйте Python и библиотеки, такие как scikit-learn.
7. Создание одностраничного приложения (SPA) на Vue.js. Разработайте SPA на Vue.js, которое позволяет пользователю искать фильмы по названию и отображать информацию о них.

8. Автоматизация тестирования с Selenium. Напишите скрипты для автоматического тестирования веб-приложения с использованием Selenium. Проверьте основные функции и пользовательский интерфейс.
9. Разработка Progressive Web App (PWA). Создайте PWA, которая позволяет пользователям просматривать статьи и сохранять их для офлайн-доступа.
10. Внедрение технологий контейнеризации с Docker. Создайте проект, который использует Docker для контейнеризации веб-приложения и базы данных. Настройте Docker Compose для управления зависимостями.

Примерные задания к проектной работе 7 семестра:

1. Использование GraphQL для API. Создайте API с использованием GraphQL, который позволяет клиентам запрашивать только необходимые данные о товарах в интернет-магазине.
2. Интеграция с облачными сервисами. Разработайте приложение, которое использует облачные сервисы, такие как AWS или Google Cloud, для хранения файлов и обработки данных.
3. Создание интерфейса на основе Figma. Разработайте веб-приложение, следуя дизайну, созданному в Figma. Обратите внимание на адаптивность и доступность интерфейса.
4. Установка и настройка системы мониторинга. Настройте систему мониторинга для вашего веб-приложения с использованием Prometheus и Grafana для визуализации метрик.
5. Разработка интернет-магазина с использованием Shopify. Создайте интернет-магазин с использованием Shopify, настроив основные функции, такие как товары, корзина и оформление заказа.
6. Внедрение технологий машинного обучения в веб-приложение. Разработайте веб-приложение, которое использует API машинного обучения (например, Google Cloud AI) для обработки изображений и предоставления рекомендаций.
7. Создание игры с использованием Phaser.js. Разработайте простую 2D-игру с использованием Phaser.js, которая включает игровую механику, уровни и систему очков.

8. Создание блога с использованием статического генератора сайтов. Разработайте блог с использованием статического генератора сайтов, такого как Gatsby или Jekyll, с возможностью добавления новых постов через Markdown.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ПКН-2 Способность разрабатывать алгоритмы и программы с использованием современных технологий программирования

1) Владеет объектно-ориентированным языком программирования на уровне знания синтаксиса и семантики, основ стандартной библиотеки

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы инкапсуляции, наследования и полиморфизма; базовой синтаксис выбранного языка программирования; принципы проектирования и методологии разработки

Уметь: применять принципы объектно-ориентированного программирования в практике; работать над групповыми проектами, включая планирование, распределение задач и совместную разработку кода; создавать тесты для проверки функциональности программ

Типовые контрольные задания

Написать программу, реализующую простую игровую логику с использованием классов и объектов.

Создать приложение для управления списком задач, используя коллекции из стандартной библиотеки.

Разработать мини-проект, включающий взаимодействие нескольких классов и демонстрирующий применение принципов ООП.

2) Использует инструментальные средства программирования (IDE, SDK, API, популярные фреймворки и библиотеки)

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: популярные SDK (Software Development Kit) и их роли в разработке приложений для конкретных платформ; основные понятия API, типы API (REST, SOAP и т.д.) и их применение в разработке; популярные фреймворки, связанные с выбранным языком программирования (Django, Flask для Python, Spring для Java, React, Angular для JavaScript); этапы разработки программного обеспечения, включая проектирование, реализацию, тестирование и деплой

Уметь: применять инструменты SDK для создания, тестирования и отладки приложений; организовывать запросы к API для получения и отправки данных, обработки ответов и интеграции функциональности в приложения; использовать системы контроля версий для отслеживания изменений в коде и управления совместной работой

Типовые контрольные задания

Создание простого веб-приложения с использованием выбранного фреймворка и интеграцией стороннего API (например, API погоды или API для работы с пользовательскими данными).

Разработка мобильного приложения с использованием соответствующего SDK и тестирование его на эмуляторе или реальном устройстве.

Выполнение проекта, в котором требуется использовать несколько библиотек для обработки данных и визуализации результатов.

3) Организует кодовую базу, ориентируется в существующем коде, демонстрирует знание общепринятых соглашений и политик в области оформления кода

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: стандарты оформления кода; инструменты для статического анализа кода

Уметь: создавать и поддерживать структуру проекта, соответствующую требованиям и стандартам; использовать линтеры и формatters для автоматической проверки и исправления оформления кода

Типовые контрольные задания

Реорганизация существующей кодовой базы с учетом лучших практик и соглашений по оформлению.

Написание документации для проекта, включая описание его архитектуры, установку и использование, а также примеры кода.

4) Проектирует текстовый, программный или графический интерфейс программной системы исходя из ее назначения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: ключевых элементов интерфейса; методологии проектирования интерфейсов (Design Thinking, Agile UX, User-Centered Design); принципы доступности (Accessibility) и инклюзивного дизайна, обеспечивающие доступность интерфейсов для всех пользователей

Уметь: использовать принципы визуального дизайна; разрабатывать графические элементы интерфейса; проектировать текстовые интерфейсы, обеспечивая их удобство и читабельность; создавать программные интерфейсы (API), учитывая требования к взаимодействию между различными компонентами системы

Типовые контрольные задания

Проведение юзабилити-тестирования существующего интерфейса и составление отчета с рекомендациями по улучшению.

Проектирование графического интерфейса мобильного приложения с учетом принципов доступности и инклюзивного дизайна.

ПКН-3 Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней

1) Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: типы структур данных и их применение; основные принципы разработки алгоритмов

Уметь: использовать простые структуры данных для хранения и обработки информации в программных проектах; адаптировать и модифицировать алгоритмы для решения специфических задач; оптимизировать существующий код, используя знания о сложностях алгоритмов и структур данных

Типовые контрольные задания

Реализация алгоритмов сортировки и поиска на конкретном примере, с анализом их эффективности на различных наборах данных.

Разработка небольшого приложения, использующего различные структуры данных для обработки информации (например, управление списком задач с использованием стека или очереди).

Написание отчета о сложности различных алгоритмов, используемых в проекте, с предложениями по улучшению производительности.

2) Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: популярные стандарты и методологии сбора требований; принципы работы с требованиями в рамках методологий Agile и Scrum; инструменты и программное обеспечение для документирования и управления требованиями

Уметь: классифицировать и приоритизировать требования в зависимости от их важности и влияния на проект; оценивать архитектурные паттерны и выбирать подходящие решения, основываясь на собранных требованиях; проводить проверки и оценку системы на соответствие функциональным и нефункциональным требованиям в процессе разработки

Типовые контрольные задания

Проведение интервью с пользователями и заинтересованными сторонами для сбора требований к новому программному продукту, с последующим формированием документации.

Разработка архитектурной схемы для информационной системы на основе собранных требований, включая описание выбранных паттернов и решений.

Анализ существующих требований и предложение улучшений или изменений на основе выявленных противоречий или недостатков.

3) Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основы одного или нескольких объектно-ориентированных языков программирования; особенности платформы или технологии, на которой будет разрабатываться система; методы и подходы к декомпозиции задач; архитектурные паттерны

Уметь: создавать интерфейсы и абстрактные классы для определения контрактов между компонентами системы; проводить юнит-тестирование и интеграционное тестирование; создавать диаграммы для визуализации структуры системы и взаимосвязей между классами

Типовые контрольные задания

Разработка небольшого программного приложения с использованием объектно-ориентированного подхода, включая создание классов, методов и интерфейсов.

Проведение декомпозиции сложной задачи, с созданием плана реализации и диаграмм, отражающих структуру системы.

Реализация и тестирование системы на выбранной платформе с использованием подходящих технологий, включая написание документации по проекту.

ПКН-5 Способность участвовать в документальном сопровождении разработки ИТ в рамках проектных групп, применять средства автоматизации управления проектами ИТ

1) Демонстрирует знание основ версионирования и управления изменениями при разработке ПО. Использует системы контроля версий для ведения совместной разработки

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы работы с веткам в системах контроля версий и стратегии их использования в командной разработке; принципы работы распределенных и централизованных систем контроля версий

Уметь: создавать и настраивать репозитории в системах контроля версий; проводить ревью кода; использовать инструменты для автоматизации управления версиями и интеграции для обеспечения CI/CD (непрерывной интеграции и доставки)

Типовые контрольные задания

Настройка репозитория в Git, создание веток для новых функций, а также совместное слияние изменений с использованием pull request.

Проведение ревью кода с последующим комментированием и предложением улучшений для повышения качества разработки.

Создание документации для проекта, включая инструкции по использованию системы контроля версий и описание процессов управления изменениями.

2) Демонстрирует знание основ тестирования программного обеспечения, умение создавать автоматизированные модульные и интеграционные тесты

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: виды тестирования; принципы модульного тестирования; популярные инструменты и фреймворки для автоматизированного тестирования

Уметь: разрабатывать интеграционные тесты для проверки взаимодействия между несколькими компонентами системы; настраивать тестовые окружения; принципы интегрирования тестов в

процесс непрерывной интеграции и доставки (CI/CD); использовать инструменты для анализа и диагностики проблем

Типовые контрольные задания

Написание набора модульных тестов для нового функционала приложения с использованием выбранного фреймворка.

Разработка интеграционных тестов для проверки взаимодействия между различными модулями системы.

Настройка автоматизированного тестирования в рамках CI/CD-процесса, включая запуск тестов при каждом коммите в репозиторий.

3) Готовит документацию к программе, коммуницирует в пределах группы разработки и за ее границами о значимых аспектах информационной системы и информационной инфраструктуры в письменной и устной форме

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы работы в команде, включая роли и ответственности участников, а также методы ведения обсуждений и собраний; методы сбора и анализа отзывов от пользователей и членов команды; принципы визуализации данных и использования диаграмм для пояснения архитектуры системы

Уметь: оформлять документацию в соответствии с установленными стандартами и требованиями; внедрять изменения и обновления в документацию на основе полученной информации

Типовые контрольные задания

Создание технической документации для нового программного продукта, включая спецификации, руководство пользователя и инструкции по установке.

Подготовка и проведение презентации о ходе проекта для заинтересованных сторон, с использованием визуальных средств для ясности изложения.

4) Демонстрирует знание жизненного цикла информационных систем, участвует в процессе разработки ПО на разных этапах

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методы мониторинга и контроля рисков на протяжении всего жизненного цикла проекта; основные критерии и метрики, по которым оценивается успешность разработки и внедрения информационной системы

Уметь: проектировать архитектуру системы и разрабатывать функциональные компоненты; анализировать требования пользователей и бизнеса, а также формализовать их в виде документации

Типовые контрольные задания

Участие в разработке подробного плана проекта, включая определение всех этапов и их сроков.

Проведение анализа требований и составление документации с описанием функциональности системы.

Разработка функционального прототипа системы и презентация его команде для получения отзывов.

Примеры практико-ориентированных заданий

1. Разработка интерфейса для ввода данных. Создайте пользовательский интерфейс в 1С для ввода данных о клиентах, включая необходимые поля и валидацию данных.
2. Настройка уведомлений о событиях. Реализуйте систему уведомлений в 1С, которая будет отправлять сообщения пользователям при изменении статуса заказа или при превышении предельного остатка на складе.
3. Создание системы прав доступа в 1С. Настройте систему прав доступа для различных ролей пользователей в вашей конфигурации, включая создание ролей и привязку прав к объектам.
4. Интеграция с внешними системами. Реализуйте интеграцию вашей конфигурации в 1С с внешней системой (например, через веб-сервисы), чтобы получать актуальные данные о курсах валют или товарах.
5. Разработка мобильного приложения на 1С:Предприятие. Создайте простое мобильное приложение для учета продаж, позволяющее пользователям вводить данные о продажах и просматривать отчеты.
6. Создание документации для конфигурации в 1С. Напишите документацию для вашей конфигурации, включая описание бизнес-процессов, функционала и инструкций по эксплуатации.

7. Проведение тестирования конфигурации в 1С. Разработайте тест-план для вашей конфигурации и проведите тестирование, включая функциональное и регрессионное тестирование.
8. Оптимизация производительности конфигурации. Проведите анализ производительности вашей конфигурации и оптимизируйте ее, исправив узкие места и улучшив запросы.
9. Реализация бизнес-процесса. Разработайте и настройте бизнес-процесс в 1С для обработки заказов, включая стадии обработки, ответственных и уведомления.
10. Внедрение Agile в проект. Разработайте план внедрения Agile-методологии в ваш проект, включая роли, церемонии и артефакты.
11. Создание постпроектного анализа. Подготовьте шаблон для постпроектного анализа, включающий оценку результатов, выявление уроков и рекомендации для будущих проектов.
12. Разработка плана по управлению конфликтами. Создайте план управления конфликтами для вашего проекта, включая методы их разрешения и стратегии предотвращения.
13. Проведение опроса среди заинтересованных сторон. Разработайте и проведите опрос среди заинтересованных сторон вашего проекта, чтобы собрать обратную связь и выявить их ожидания.
14. Создание карты процессов проекта. Задание: Разработайте карту процессов для вашего проекта, включая основные этапы и взаимодействия между ними.
15. Анализ альтернативных подходов к проекту. Задание: Проведите анализ альтернативных решений для вашего проекта и оцените их преимущества и недостатки.

Примерные вопросы для подготовки к зачету, зачету, зачету, зачету, зачету

3 семестр

1. Понятие и основные параметры проекта. Цель и стратегия проекта. Результат проекта.
2. Классификация проектов.
3. Проектный цикл. Структуризация проектов.

4. Участники проектов.
5. Окружающая среда проекта.
6. Сущность и принципы управления проектами. История развития управления проектами.
7. Функции и подсистемы управления проектами. Методы управления проектами.
8. Разработка концепции проекта. Формирование идеи проекта. Предварительные исследования по проекту.
9. Проектный анализ. Оценка реализуемости проекта.
10. Техничко-экономическое обоснование проекта.
11. Бизнес-план проекта.
12. Создание коммуникационной системы проекта.
13. Принципы построения организационных структур управления проектами. Последовательность разработки и создания организационных структур управления проектами.
14. Современные средства организационного моделирования проектов.
15. Источники финансирования. Организационные формы финансирования. Организация проектного финансирования.

4 семестр

1. Что такое проект и какие его основные характеристики?
2. Какие этапы включает в себя жизненный цикл проекта?
3. Что такое устав проекта и какие элементы он содержит?
4. Каковы основные задачи менеджера проекта?
5. В чем разница между проектом и операционной деятельностью?
6. Что такое заинтересованные стороны проекта и как их идентифицировать?
7. Каковы основные методы оценки рисков в проекте?
8. Что такое SMART-цели и как их применять в проектном управлении?
9. Как составить рабочий план проекта?

10. Что такое диаграмма Ганта и как она используется?

5 семестр

1. Какие методы существуют для оценки временных затрат на задачи проекта?

2. Что такое критический путь и как его определить?

3. Как составить бюджет проекта и какие элементы он включает?

4. Как управлять ресурсами в проекте?

5. Каковы основные принципы Agile-методологии?

6. В чем разница между методами работы по водопаду и Agile?

7. Как проводить мониторинг и контроль выполнения проекта?

8. Что такое управление изменениями и зачем оно нужно?

9. Какие существуют техники оценки и управления рисками?

6 семестр

1. Как учитывать культурные различия при управлении международными проектами?

2. Что такое проектный офис и какова его роль в организации?

3. Как управлять проектами в условиях неопределенности?

4. Какие существуют методы оценки стоимости проекта?

5. Как осуществлять управление качеством на всех этапах проекта?

6. Как разработать бизнес-кейс для проекта?

7. Какие инструменты и программное обеспечение могут быть полезны для управления проектами?

8. Как проводить SWOT-анализ проекта?

9. Какие методы оценки эффективности проекта существуют?

10. Как реализовать систему управления проектами в организации?

7 семестр

1. Какова роль коммуникации в управлении проектом?

2. Какие виды отчетности существуют в проектном управлении?
3. Как организовать эффективные встречи команды проекта?
4. Как управлять конфликтами в команде проекта?
5. Какова важность обратной связи в проекте?
6. Как проводить анализ успешности проекта после его завершения?
7. Что такое KPI и как они применяются в проектном управлении?
8. Каковы основные методы тестирования и контроля качества в проектах?
9. Как проводить постпроектный анализ и какие элементы он включает?
10. Как использовать методологии управления проектами?

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Куликов, А. А. Инструментальное программное обеспечение разработки и проектирования информационных систем [Электронный ресурс] / Куликов А. А., Матчин В. Т., Синицын А. В., Литвинов В. В. Москва : РТУ МИРЭА, 2022 263 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/311003>. [БИК ID: RU-LAN-BOOK-311003]
2. Кушнер, М. А. Проектный менеджмент [Электронный ресурс] : учебное пособие / Кушнер М. А., Кушнер А. А., Дубинина Н. А., Тараскина Ю. В. Астрахань : АГТУ, 2023 100 с. Допущено редакционно-издательским советом Астраханского государственного технического университета в качестве учебного пособия для обучающихся по программам высшего образования по направлению подготовки 38.03.02 Менеджмент Книга из коллекции АГТУ - Экономика и менеджмент <https://e.lanbook.com/book/411950> ISBN 978-5-89154-753-7. [БИК ID: RU-LAN-BOOK-411950]
3. Баланов, А. Н. Внедрение методологий в IT: Agile, Scrum и другие [Электронный ресурс] : учебное пособие для спо / Баланов А. Н. Санкт-Петербург : Лань, 2024 68 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/401126> ISBN 978-5-507-48920-6. [БИК ID: RU-LAN-BOOK-401126]

Дополнительная литература:

1. Проектный практикум : учебное пособие / Горожанина Е. И. Ч. 1 : Проектный практикум. Часть 1 : учебное пособие . Ч. 1 / Горожанина Е. И. Самара : ПГУТИ, 2022 136 с. Книга из коллекции ПГУТИ - Экономика и менеджмент <https://e.lanbook.com/book/411584>. [БИК ID: RU-LAN-BOOK-411584]
2. Смирнов, М. В. Проектирование и разработка информационных систем и бизнес-приложений: Методические указания [Электронный ресурс] / Смирнов М. В., Исаев Р. А., Толмасов Р. С. Москва : РТУ МИРЭА, 2020 44 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/163878>. [БИК ID: RU-LAN-BOOK-163878]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
(<http://library.fa.ru/files/elibfa.pdf>)
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
4. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
5. •Электронно-библиотечная система Znanium <http://www.znanium.ru/>

10. Методические указания для обучающихся по освоению дисциплины

Самостоятельная работа студента направлена на подготовку к учебным занятиям и на развитие знаний, умений и навыков, предусмотренных программой дисциплины.

В соответствии с учебным планом дисциплина предусматривает практические занятия, а также выполнение и защиту проектной работы. Успешное изучение дисциплины требует посещения всех видов занятий, выполнение заданий преподавателя и ознакомления с основной и дополнительной литературой. В зависимости от мероприятий, предусмотренных учебным планом и разделом 4, данной программы, студент выбирает методические указания для самостоятельной работы из приведённых ниже.

Практические занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности студентов по изучаемой дисциплине. При подготовке к практическому занятию студенты имеют возможность воспользоваться консультациями преподавателя. В ходе данных занятий необходимо давать конкретные, четкие ответы по существу вопросов;

на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин), не имеющим письменного решения задач или не подготовившихся к данному практическому занятию, рекомендуется не позже, чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, изученную на занятии.

Методические указания, необходимые для изучения и прохождения дисциплины приведены в составе образовательной программы.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Python 3.8
2. 1С: Предприятие 8.3
3. Среда разработки RStudio
4. Adobe Acrobat
5. Figma
6. Kaspersky
7. QS Analytics
8. Windows
9. Текстовый редактор
10. Docker
11. Реляционная СУБД
12. Anaconda
13. Статистический программный продукт STATISTICA
14. Power BI
15. Онлайн-приложение для бизнес-аналитики Yandex DataLens

Современные профессиональные базы данных и информационные справочные системы:

1. Язык программирования Python 3. <https://pythonworld.ru/>
2. Yandex DataLens
3. СУБД MongoDB
4. Пакет «STATISTICA»
5. Система обнаружения заимствований «Антиплагиат»: <https://fa.antiplagiat.ru>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. Помещение для самостоятельной работы обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде Университета.

2. Компьютерный класс для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)

3. Библиотека, читальный зал