

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финуниверситет)**

Владикавказский филиал Финуниверситета

Кафедра «Корпоративные инфокоммуникационные системы»

УТВЕРЖДАЮ

Директор филиала

Т.А. Хубаев

2026 г.



М.А. Ковалева

Современные технологии программирования

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

09.03.04 Программная инженерия,

ОП «Технологии разработки программного обеспечения»

*Рекомендовано Ученым советом Владикавказского филиала
Финансового университета*

(протокол от « 15 » апреля 2026 г. № 30)

*Одобрено на заседании кафедры «Корпоративные инфокоммуникационные
системы»*

(протокол от « 10 » апреля 2026 г. № 8)

Владикавказ 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	5
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	5
5.1. Содержание дисциплины	5
5.2. Учебно-тематический план	10
5.3. Содержание семинаров, практических занятий.....	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.....	32
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	47
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	48
10. Методические указания для обучающихся по освоению дисциплины.....	48
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	59
11.1 Комплект лицензионного программного обеспечения	59
11.2 Современные профессиональные базы данных, и информационные справочные системы	59
11.3 Сертифицированные программные и аппаратные средства защиты информации	59
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	59

1. Наименование дисциплины

Дисциплина «Современные технологии программирования».

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
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	1. Демонстрирует знания основных программных продуктов, используемых для решения задач профессиональной деятельности, в том числе, отечественного производства.	Знать: современные интегрированные среды разработки, используемые для решения задач профессиональной деятельности. Уметь: применять современные интегрированные среды при разработке и реализации программных продуктов.
		2. Применяет готовые инструментальные средства для решения задач профессиональной деятельности, проводит квалифицированную их оценку и обосновывает свой выбор.	Знать: методы инструментальной разработки и возможности их применения при решении профессиональных задач; методы оценки существующих инструментальных средств. Уметь: применять методы инструментальной разработки и оценки существующих инструментальных средств, а также обосновывать их при реализации программных продуктов, в том числе и информационных систем.
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а	1. Демонстрирует знания основных стандартов ведения технической документации, как отечественных, так и зарубежных.	Знать: основные стандарты ведения технической, как отечественной, так и зарубежной технической документации. Уметь: составлять и вести,

	также технической документации, связанной с профессиональной деятельностью		как отечественную, так и зарубежную техническую документацию.
		2. Создает комплект программной и проектной документации к разрабатываемой информационной системе.	Знать: методы создания комплекта программной и проектной документации к разрабатываемой информационной системе. Уметь: создавать комплект программной и проектной документации к разрабатываемой информационной системе.
		3. Критически анализирует программную и проектную документацию, строит на ее основе логические выводы.	Знать: методы критического анализа программной и проектной документации. Уметь: критически анализировать программную и проектную документацию; строить на основе программной и проектной документации логические выводы.
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	1. Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования.	Знать: методы разработки и алгоритмы решения простых информационных задач Уметь: разрабатывать и применять алгоритмы решения простых информационных задач
		2. Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ.	Знать: методы анализа алгоритмов в части производительности и оптимальности; методы создания рекомендаций для оптимизации алгоритмов программ. Уметь: анализировать алгоритмы в части производительности, оптимальности, а также вырабатывать рекомендации для оптимизации алгоритмов программ.
		3. Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев.	Знать: методы черного и белого ящика, составляет набор тестовых случаев, а также методы ручного и автоматизированного тестирования, в том числе и создания UNIT-тестов.

			Уметь: проводить ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев.
--	--	--	--

3. Место дисциплины в структуре образовательной программы

Дисциплина «Современные технологии программирования» является дисциплиной общефакультетского (предпрофильного) цикла части, формируемой участниками образовательных отношений, учебного плана образовательной программы «Технологии разработки программного обеспечения» по направлению подготовки 09.03.04 Программная инженерия, профиль «Технологии разработки программного обеспечения».

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

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 4 (в часах)	Семестр 5 (в часах)
Общая трудоемкость дисциплины	6/216	108	108
Контактная работа- Аудиторные занятия	100	50	50
<i>Лекции</i>	32	16	16
<i>Семинары, практические занятия</i>	68	34	34
Самостоятельная работа	116	58	58
Вид текущего контроля	Проектная работа, курсовой проект	Проектная работа	Курсовой проект
Вид промежуточной аттестации	Зачет, экзамен	Зачет	Экзамен

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. Обзор платформы и языка программирования Java

Общая информация о языке программирования Java. История языка программирования, его связь с другими языками программирования, распространенность Java и основные сферы его применения. Знакомство с первыми примерами кода на Java. Базовая информация о языке программирования Java. Базовая информация о синтаксисе. Базовая информация о типизации. Информация о примитивных типах данных. Математические операции над числовыми типами данных. Преобразование типов данных. Переменные и специфика их объявления. Именованные переменные. Основная информация о Java Development Kit. Основная информация о Java Runtime Environment. Информация о JIT-компиляции. Виртуальная машина Java. Сборщик мусора. Интегрированная среда разработки Java.

Тема 2. Основные языковые конструкции Java. Коллекции

Строковый тип данных языка Java. StringBuilder, StringBuffer. Идентификаторы, ключевые слова, литералы, операторы, разделители. Управляющие конструкции в Java. Булев тип: объявление и операции. Операции сравнения в Java. Условные операторы в Java. Оператор switch-case в Java. Массивы, особенности инициализации. Циклы в Java: while, do-while, for, for each. Специфика циклов в Java. Описание Java Collection. Классификация коллекций java: Collection, Map. Основные методы интерфейса Collection. Interface Iterator<E>, ListIterator<E>. Списки (Interface List<E>), имплементации List: ArrayList, LinkedList. Интерфейс Set. Реализации интерфейса Set: HashSet, TreeSet. Очереди java, интерфейс Queue, основные методы. Интерфейс Deque.

Тема 3. Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ.

Базовые принципы объектно-ориентированного программирования: наследование, инкапсуляция, полиморфизм. Объектная модель Java: пакеты, классы и объекты. Спецификации классов (модификаторы доступа).

Отношения между классами. Конструкторы, методы, переменные и поля классов. Объекты и объектные переменные класса. Ключевое слово `final`. Статические поля и методы классов. Пакеты в Java. Импорт пакетов и классов. Вложенные и внутренние классы. Полиморфизм. Перегрузка методов и конструкторов. Ключевое слово `this`. Классы `java.util`. Суперклассы и подклассы. Ключевое слово `super`. Наследование и доступ к членам класса. Иерархии наследования. Восходящее и нисходящее преобразование типов. Ключевое слово `instanceof`. Переопределение методов. Абстрактные классы и методы. Типизированные классы и методы (Generic Class, Generic Method). Наследование Generics. Ограничения, налагаемые на параметры. Совместимость на уровне байт-кода. Реализация интерфейсов. Свойства интерфейсов. Переменные интерфейсов. Множественная реализация интерфейсов. Наследование интерфейсов. Методы интерфейсов: по умолчанию, статические. Обработка исключений. Классификация исключений. Классы `Throwable`, `Exception`, `RuntimeException`, `IOException`.

Тема 4. Потокковая организация системы ввода-вывода. Многопоточность.

Чтение и запись файлов с использованием байтовых потоков. Заккрытие файлов. Файлы с произвольным доступом. Файловый ввод-вывод с использованием символьных потоков. Классы `FileWriter` и `FileReader`. Потокковая организация системы ввода-вывода Java. Абстрактные классы `InputStream` и `OutputStream`. Многопоточное программирование, модель потока в Java. Многопоточность в одно- и много процессорных системах. Класс `Thread` и интерфейс `Runnable`: основные методы, создание, приостановка, прерывание, завершение потока. Группы потоков. Жизненный цикл потока. Фоновые задачи. Управление потоками и приоритеты потоков. Синхронизация.

Тема 5. Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование.

Синтаксис. Переменные. Типы данных. Область видимости: глобальные и локальные переменные. Константы. Основные операции. Функции, их объявление и использование. Указатели. Условный оператор и обработка ошибок. Массивы и срезы. Цикл for. Нулевое значение массива. Мапы (отображения). Встроенная функция delete(). Структуры. Методы структур. Указатели на структуру. Указатели в роли ресивера. Кастомные типы. Интерфейсы. Хранилище. Пустые интерфейсы.

Тема 6. GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты.

Конкурентность. Параллелизм. Процессы. Адресное пространство. Потоки. Конкурентность в Go. Потоки и Горутины. Анонимные горутины. Каналы, как механизм коммуникации между горутинами. Чтение из канала в цикле и deadlock. Не блокирующее чтение. Система модулей. Создание собственного модуля. Импортирование зависимостей. Работа с пакетами.

Тема 7. Микросервисная архитектура. Оркестратор. Backend-For-Frontend.

Микросервис. Монолит. Независимые компоненты. Децентрализация данных. Распределенная архитектура. Оркестратор, как связующее звено в микросервисной архитектуре. Концепция Backend For Frontend.

Тема 8. Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных языках программирования. Механизм миграций.

Архитектура API JDBC (Java DataBase Connectivity), основные компоненты, модели обработки. Преимущества и недостатки применения JDBC. Драйверы JDBC. Установление соединения, классы DriverManager и DataSource, интерфейс Connection. Типы данных SQL и JDBC. Классы Statement, PreparedStatement. Запросы к базе данных. Добавление, удаление, изменение информации в базе данных. Обработка результатов выполнения запроса к базе данных, интерфейс ResultSet. Заккрытие соединения. Основы Java Persistence API: Entity, встраиваемые классы, управление связями,

стратегии наследования маппинга. Основы Hibernate Framework. Основы объектно-реляционного отображения. Основы языка запросов HQL. Отличия Entity в Hibernate и Java Persistence API. Основы GORM. Ассоциации. Хуки. Нетерпеливая загрузка. Транзакции. Вложенные транзакции. Откат. Контекст, режим подготовленного оператора. SQL Builder. Механизм миграций GORM. Основы SQLAlchemy для языка программирования Python. Запросы и relations. Query builder. Alembic для миграция в Python. Psycopg2. Asyncpgsa. MetaData, Dialect, Engine.

Тема 9. Системы автоматической сборки пакетов. Контейнеризация. Docker.

Автоматизация сборки. Компиляция, управление зависимостями, автоматическое тестирование, пакетирование приложений. Основные особенности Maven. Структура языка разметки при помощи XML. Основные особенности Gradle. Структура языка разметки при помощи Groovy. Сравнение систем автоматической сборки. Контейнеры. Развертывание контейнеров. Структура языка разметки при помощи YAML. Application.YAML. Docker. DockerHub.

Тема 10. Spring Framework. Архитектура, основные сведения.

Основные фреймворки и задачи, решаемые Spring. Внедрение зависимостей. Интерфейс ApplicationContext. Inversion of Control – инверсия управления. Dependency Injection – внедрение зависимости между объектами приложения. Внедрение зависимости при помощи конструктора класса и методов set. Конфигурирование метаданных: xml конфигурация, Java-аннотация, Java-код. Объект Bean. Область видимости объектов Bean: singleton, prototype, request, session, application. Жизненный цикл объекта Bean. Спецификация, применение init- и destroy-методов. Аннотации @Component, @Autowired. Основы интерфейса Servlet.

Тема 11. RESTful API приложения.

Архитектура REST (Representational State Transfer — «передача состояния представления») и RESTful API. Spring MVC (Model-View-

Controller), общая схема работы. Класс DispatcherServlet. Аннотации @Controller, @Component, @ComponentScan. Маппинг (определение соответствия): аннотации @GetMapping, @PostMapping, @RequestMapping. Обработка HTTP-запросов: интерфейсы HttpServletRequest, Model. Классификация функций по манипуляции данными CRUD (create, read, update, delete). Взаимосвязь CRUD и REST. Связь с базой данных на основе паттерна DAO (Data Access Object). Применение класса JdbcTemplate.

Тема 12. GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения.

Рутинг, маршрутизация, рендеринг. Middleware. Обработчики запросов. Привязка данных запросов. Ограничения фреймворка Gin. Функции. Тесты. JSONiter. Развертывание. Валидаторы. FastAPI. Path-параметры. Query-параметры. Связь FastAPI и SQLAlchemy. Тело запроса. Query-параметры и валидация строк. Параметры Cookie. Header-параметры. Модель ответа. Request Files. Обработка ошибок. Конфигурация операций пути. Настройка авторизации. Middleware. CORS (Cross-Origin Resource Sharing). Фоновые задачи. Статические файлы. Тестирование

Тема 13. Логирование. Документирование.

Swagger. Развертывание swagger. Синтаксис языка для организации swagger без кода. Генерация swagger на основе Spring Framework, Gin, FastAPI. Структура логирования Apache. Библиотека logging для языка программирования Python. Пакет slf4j для языка программирования Java. Пакет log для языка программирования Go.

5.2. Учебно-тематический план

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоятельна я работа	
			Общая, в т.ч.:	Лекции и	Семинары, практи		
1	Тема 1. Обзор платформы и	16	8	2	6	8	Опрос, решение задач

	языка программирования Java						
2	Тема 2. Основные языковые конструкции Java. Коллекции	16	8	2	6	8	Опрос, решение задач
3	Тема 3. Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ	16	8	2	6	8	Опрос, решение задач
4	Тема 4. Поточковая организация системы ввода-вывода. Многопоточность	16	8	2	6	8	Опрос, решение задач
5	Тема 5. Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование	16	8	2	6	8	Опрос, решение задач, защита проектной работы
6	Тема 6. GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты	16	8	2	6	8	Опрос, решение задач
7	Тема 7. Микросервисная архитектура. Оркестратор. Backend-For-Frontend	16	8	2	6	8	Опрос, решение задач
8	Тема 8. Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных	16	8	2	6	8	Опрос, решение задач

	языках программирования . Механизм миграций						
9	Тема 9. Системы автоматической сборки пакетов. Контейнеризация. Docker	14	6	2	4	8	Опрос, решение задач
10	Тема 10. Spring Framework. Архитектура, основные сведения	18	8	4	4	10	Опрос, решение задач
11	Тема 11. RESTful API приложения	18	8	4	4	12	Опрос, решение задач
12	Тема 12. GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения	20	8	4	4	12	Опрос, решение задач
13	Тема 13. Логирование. Документирование	18	6	2	4	12	Опрос, решение задач, защита курсового проекта
В целом по дисциплине		216	100	32	68	116	Согласно учебному плану: проектная работа, курсовой проект
Итого в %		100	46,3	32	68	53,7	

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятия
Тема 1. Обзор платформы и языка программирования Java	Комплект разработчика (Java development kit), среда выполнения Java (Java Runtime Environment). Интегрированная среда разработки (Integrated development Environment) IntelliJ Idea. Установка. Создание нового проекта, системы сборки проектов Maven. Структура проекта. Интерфейс IDE IntelliJ Idea. Создание исполняемого файла. Использование инструментов командной строки. Примитивные и ссылочные типы данных, литералы. Операции. Преобразование типов. Переменные: объявление, инициализация.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 2. Основные языковые	Консольный ввод-вывод данных, команды ввода-вывода. Переменные и основные типы	Интерактивная форма: опрос,

конструкции Java. Коллекции	переменных. Объявление и инициализация переменных. Присвоение данных между разными типами переменных. Инициализация переменных. Специальные операторы. Операторы «вычисление остатка» и «вычисление частного». Логический тип (булевы значения). Ветвление (конструкция «if – else»). Класс Math. Циклы for и while. Массивы и методы. Строковые и символьные переменные. Классы String, StringBuilder, StringBuffer. Интерфейсы Collection, Set, List, Map, Iterator. Каркас коллекций. Представления и оболочки. Унаследованные коллекции. Абстрактные классы, используемые при работе с коллекциями. Интерфейс Comparator. Применение ArrayList. TreeMap. Применение очередей. Алгоритмы: сортировка, двоичный поиск.	решение задач с последующим коллективным обсуждением их результатов
Тема 3. Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ	Объекты и классы. Состояние и поведение. Построение классов в соответствии с принципами инкапсуляции, полиморфизма и наследование. Перегрузка конструкторов, методов. Области видимости, модификаторы доступа. Применение ключевого слова static. Объявление классов. Метод main. Отношения между классами. Имена классов, импорт. Инициализаторы. Преобразование ссылочных типов. Ключевые слова this и super. Обобщенные методы и конструкторы. Применение ключевого слова final к переменным экземпляра класса. Объявление, реализация и применение интерфейсов. Наследование интерфейсов. Методы интерфейсов: по умолчанию, статические. Ошибки при работе программы. Исключения (Exceptions). Конструкция try-catch-finally. Использование оператора throw. Отладка программ в среде программирования IDE IntelliJ IDEA. Запуск отладчика, точки останова.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 4. Потоквая организация системы ввода-вывода. Многопоточность	Консольный ввод, запись консольного вывода. Классы-реализации потоков данных. Классы ByteArrayInputStream и ByteArrayOutputStream. Классы FileInputStream и FileOutputStream. StringBufferInputStream. Классы FilterInputStreeam и FilterOutputStream. BufferedInputStream и BufferedOutputStream. DataInputStream и DataOutputStream. Класс Thread и интерфейс Runnable. Жизненный цикл потока. Фоновые задачи.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов

	Управление потоками и приоритеты потоков. Синхронизация. Приостановка, возобновление и остановка потока.	
Тема 5. Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование	Переменные и константы. Типизация. Перевод типов данных. Условная конструкция. Построение и использование функций. Интерфейсы. Структуры и методы. Цикл. Условная конструкция для управления ошибками.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов, защита проектной работы
Тема 6. GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты	Горутин. Канал. Конкурентность. Параллелизм. Канал взаимодействия горутинов.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 7. Микросервисная архитектура. Оркестратор. Backend-For-Frontend	Построение микросервисов. Создание оркестратора. Создание нескольких микросервисов и взаимосвязь через оркестратор.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 8. Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных языках программирования. Механизм миграций	Установление соединения с системами управления баз данных MySQL и PostgreSQL. Драйверы JDBC. Установление соединения, классы DriverManager и DataSource, интерфейс Connection. Классы Statement, PreparedStatement. Запросы к базе данных. Добавление, удаление, изменение информации в базе данных. Обработка результатов выполнения запроса к базе данных, интерфейс ResultSet. Закрытие соединения. Драйверы JPA. Hibernate framework. Классы Entity, встраиваемые классы, наследование маппингов, fetch стратегии. Жизненный цикл Entity, EntityManager. SQLAlchemy. Жизненный цикл модели. Alembic. Миграция. Создание миграций. Настройка файла миграций. GORM. Создание модели. Миграция. Создание миграций. Настройка файла миграций.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 9. Системы	Сборка проекта при помощи Maven. Основы	Интерактивная

автоматической сборки пакетов. Контейнеризация. Docker	синтаксиса Maven. Сборка проекта при помощи Gradle. Синтаксис Groovy. Контейнер. Контейнеризация. Docker.	форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 10. Spring Framework. Архитектура, основные сведения	Конфигурация приложения при помощи xml файлов, java-аннотации. Внедрение зависимостей на основе конструкторов и set-методов. Спецификация, применение init- и destroy-методов. Аннотации @Component, @Autowired. Автоматическое связывание свойств компонентов.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 11. RESTful API приложения	Использование классов поддержки DAO. Класс DispatcherServlet. Аннотации @Controller, @Component, @ComponentScan. Маппинг (определение соответствия): аннотации @GetMapping, @PostMapping, @RequestMapping. Обработка HTTP-запросов: интерфейсы HttpServletRequest, Model.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 12. GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения	Интеграция GIN. Архитектура. ShouldBindUri(). C.JSON. ListenAndServe() FastAPI. Typing Union. @app. @app.get, @app.post().	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 13. Логирование. Документирование	Структура логирования Apache. Архитектура slg4j. Логирование в файл. Оформление структуры. Пакет logging. Логирование в файл. Оформление структуры. Пакет log. Log.new(). Логирование в файл. Оформление структуры.	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов, защита курсового проекта

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
Тема 1. Обзор платформы и языка программирования	Установка IDE IntelliJ Idea. Переменные среды JDK. Регистрация github, создание проекта для выполнения семинарских занятий.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и

я Java		анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 2. Основные языковые конструкции Java. Коллекции	Создание блоков кода. Стилевое оформление программного кода. Ключевые слова. Укороченные логические операции. Многоступенчатые конструкции «if – else – if». Оператор Switch. Цикл без тела. Вложенные циклы.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 3. Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ	Поиск пакета и переменная среды CLASSPATH. Защищенные члены классов. Класс Object. Многоуровневая иерархия классов. Передача объектов методам. Инициализация полей по умолчанию. Вызов одного конструктора из другого. Интерфейсы и абстрактные классы. Разрешение конфликтов в методах по умолчанию. Лямбда-выражения. Генерация исключений в лямбда-выражениях. Переопределение методов и исключения.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 4. Потоквая организация системы ввода-вывода. Многопоточность	Связные списки. Хэш-множества. Очереди по приоритету. Древовидные множества. Взаимные преобразования списков и множеств. Стеки. Битовые множества. Классы PipedInputStream и PipedOutputStream. SequenceInputStream. LineNumberInputStream. PushBackInputStream. PrintStream. Синхронизированные блоки. Параллельное программирование.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 5. Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-	Установка GoLang. Использование Структур. Вызов методов структур. Использование указателей при передаче массивов в функции.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение

ориентированное программирование		освоение, выполнение проектной работы
Тема 6. GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты	Создание горутина. Создания канала. Создание модуля. Объединение модулей в пакеты. Параллелизм.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 7. Микросервисная архитектура. Оркестратор. Backend-For-Frontend	Построение backend-for-fronted архитектуры. Построение оркестраторной архитектуры.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 8. Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных языках программирования. Механизм миграций	Исключения (SQL Exception). Потoki в JDBC. Основные этапы работы с базой данных при использовании JDBC. Уровни изоляции транзакций в JDBC. Хранимая процедура.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 9. Системы автоматической сборки пакетов. Контейнеризация. Docker	Сборка пакета Maven. Сборка пакета Gradle. Конфигурация Groove. Конфигурация application.yaml. Запуск Docker.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 10. Spring Framework. Архитектура, основные	Аспектно-ориентированный Spring. Выбор точек сопряжения в описаниях срезов. Объявление аспектов в XML. Аннотирование аспектов. Внедрение	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и

сведения	аспектов AspectJ.	анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 11. RESTful API приложения	Построение REST API приложения при помощи SPRING.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 12. GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения	Построение сервиса при помощи GIN. Построение сервиса при помощи FastAPI. FastAPI Secure. OAuth 2.0 авторизации.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 13. Логирование. Документирование	Формирование логирования при помощи соответствующих библиотек.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение курсовой работы

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

Примерный перечень тем для подготовки к опросу

1. Базовая информация о языке программирования Java.
2. Базовая информация о синтаксисе.

3. Базовая информация о типизации. Информация о примитивных типах данных.

4. Математические операции над числовыми типами данных.

5. Преобразование типов данных.

6. Переменные и специфика их объявления.

7. Основная информация о Java Development Kit.

8. Основная информация о Java Runtime Environment.

9. Информация о JIT-компиляции. Виртуальная машина Java.

10. Интегрированная среда разработки Java.

11. Строковый тип данных языка Java. StringBuilder, StringBuffer.

Идентификаторы, ключевые слова, литералы, операторы, разделители.

12. Управляющие конструкции в Java. Булев тип: объявление и операции. Операции сравнения в Java. Условные операторы в Java.

13. Массивы, особенности инициализации.

14. Циклы в Java: while, do-while, for, for each. Специфика циклов в Java.

15. Базовые принципы объектно-ориентированного программирования: наследование, инкапсуляция, полиморфизм.

16. Объектная модель Java: пакеты, классы и объекты.

17. Спецификации классов (модификаторы доступа).

18. Пакеты в Java. Импорт пакетов и классов. Вложенные и внутренние классы. Полиморфизм. Перегрузка методов и конструкторов.

19. Реализация интерфейсов. Свойства интерфейсов. Переменные интерфейсов.

20. Множественная реализация интерфейсов. Наследование интерфейсов. Методы интерфейсов:

21. Чтение и запись файлов с использованием байтовых потоков. Закрытие файлов.

22. Файлы с произвольным доступом. Файловый ввод-вывод с использованием символьных потоков.

23. Многопоточное программирование, модель потока в Java.
24. Многопоточность в одно- и много процессорных системах.
25. Управление потоками и приоритеты потоков. Синхронизация.
26. Функции, интерфейсы, объектно-ориентированное программирование.
27. Синтаксис. Переменные. Типы данных. Область видимости: глобальные и локальные переменные. Константы. Основные операции.
28. Структуры. Методы структур. Указатели на структуру. Указатели в роли ресивера. Кастомные типы.
29. Интерфейсы. Хранилище. Пустые интерфейсы.
30. Конкурентность. Параллелизм. Процессы. Адресное пространство. Потоки.
31. Потоки и Горутины. Анонимные горутины.
32. Каналы, как механизм коммуникации между горутинами.
33. Система модулей. Создание собственного модуля. Импортирование зависимостей. Работа с пакетами.
34. Микросервис. Монолит. Независимые компоненты. Децентрализация данных. Распределенная архитектура.
35. Оркестратор, как связующее звено в микросервисной архитектуре.
36. Концепция Backend For Frontend.
37. Архитектура API JDBC (Java DataBase Connectivity), основные компоненты, модели обработки.
38. Преимущества и недостатки применения JDBC.
39. Типы данных SQL и JDBC.
40. Запросы к базе данных. Добавление, удаление, изменение информации в базе данных.
41. Обработка результатов выполнения запроса к базе данных. Закрытие соединения.
42. Основы Java Persistence API: Entity, встраиваемые классы,

управление связями, стратегии наследования маппинга.

43. Основы Hibernate Framework.

44. Основы объектно-реляционного отображения. Основы языка запросов HQL.

45. Основы GORM. Ассоциации. Хуки. Нетерпеливая загрузка. Транзакции. Вложенные транзакции. Откат. Контекст, режим подготовленного оператора. SQL Builder. Механизм миграций GORM.

46. Основы SQLAlchemy для языка программирования Python. Запросы и relations. Query builder.

47. Alembic для миграция в Python. Psycopg2. Asyncpgsa. MetaData, Dialect, Engine.

48. Автоматизация сборки.

49. Компиляция, управление зависимостями, автоматическое тестирование, пакетирование приложений.

50. Основные особенности Maven.

51. Структура языка разметки при помощи XML.

52. Основные особенности Gradle.

53. Структура языка разметки при помощи Groovy.

54. Контейнеры. Развертывание контейнеров. Структура языка разметки

55. Основные фреймворки и задачи, решаемые Spring.

56. Конфигурирование метаданных: xml конфигурация, Java-аннотация, Java-код.

57. Жизненный цикл объекта Bean.

58. Спецификация, применение init- и destroy-методов.

59. Основы интерфейса Servlet.

60. Архитектура REST (Representational State Transfer — «передача состояния представления») и RESTful API.

61. Spring MVC (Model-View-Controller), общая схема работы. Класс DispatcherServlet. Аннотации @Controller, @Component, @ComponentScan.

62. Маппинг (определение соответствия): аннотации @GetMapping, @PostMapping, @RequestMapping. Обработка HTTP-запросов: интерфейсы HttpServletRequest, Model.

63. Классификация функций по манипуляции данными CRUD (create, read, up-date, delete).

64. Рутинг, маршрутизация, рендеринг. Обработчики запросов. Привязка данных запросов. Функции. Тесты. Развертывание. Валидаторы.

65. FastAPI. Path-параметры. Query-параметры. Связь FastAPI и SQLAlchemy. Тело запроса. Обработка ошибок. Конфигурация операций пути. Настройка авторизации. Фоновые задачи. Статические файлы. Тестирование

66. Логирование. Документирование.

Примеры задач

Задача 1. Составление технической документации с JavaDoc.

Составить техническую документацию на разработанную систему с использованием JavaDoc:

1. Добавить JavaDoc-комментарии ко всем классам, интерфейсам, полям и методам, включая:
 - описание назначения;
 - параметры (@param);
 - возвращаемое значение (@return);
 - исключения (@throws).
2. Сгенерировать HTML-документацию с помощью утилиты javadoc.
3. Проверить полноту и корректность сгенерированной документации.
4. Включить в документацию описание основных классов, их взаимосвязей и ключевых методов.

Задача 2. Составление проектной документации.

Подготовить проектную документацию для системы:

1. Техническое задание:
 - цель и назначение системы;
 - функциональные и нефункциональные требования;
 - ограничения и допущения.
2. Архитектурная документация:
 - схема компонентов системы;
 - диаграмма развёртывания;
 - описание используемых технологий и их взаимодействия.
3. Руководство пользователя:
 - описание основных функций системы;
 - сценарии использования (Use Cases);
 - инструкции по работе с интерфейсом.
4. Руководство администратора:
 - установка и настройка системы;
 - управление пользователями и правами доступа;
 - мониторинг и резервное копирование.

Задача 3. Анализ и исправление документации.

Проанализировать программную и проектную документацию, выявить ошибки и исправить их:

1. Проверить JavaDoc-комментарии на:
 - полноту (все ли элементы задокументированы);
 - актуальность (соответствуют ли комментарии реальному коду);
 - корректность (правильность использования тегов, грамматика).
2. Проанализировать проектную документацию на:
 - наличие всех обязательных разделов;
 - ясность и однозначность формулировок;
 - соответствие архитектуры и требований реальному решению.
3. Выявить и исправить следующие типы ошибок:
 - противоречия между разделами документации;

- устаревшую информацию;
- отсутствие описания ключевых функций;
- некорректные диаграммы и схемы.

4. Подготовить отчёт о найденных ошибках и внесённых исправлениях.

Задание 4. Разработать конвертер температур (Цельсий ↔ Фаренгейт ↔ Кельвин) с графическим интерфейсом на JavaFX.

Реализовать:

- поля ввода/вывода для каждой шкалы;
- кнопки конвертации между всеми парами шкал;
- валидацию ввода (только числа);
- обработку ошибок (например, температура ниже абсолютного нуля).

Выполнить документирование на основе Javadoc. Написать тест к этому приложению на основе JUnit 5, проверяющий корректность вычислений.

Задание 5. Создать приложение «Список дел» на JavaFX.

Реализовать:

- поле для ввода новой задачи;
- кнопка «Добавить»;
- список задач с чекбоксами для отметки выполнения;
- кнопка «Удалить выполненные»;
- сохранение списка в файл при закрытии приложения и загрузка при запуске.

Выполнить документирование на основе Javadoc.

Задание 6. Разработать калькулятор ИМТ (индекс массы тела) с графическим интерфейсом JavaFX.

Реализовать:

- ввод роста (в см) и веса (в кг);
- вычисление ИМТ и вывод категории (недостаток веса, норма, избыток, ожирение);

- визуальную индикацию результата (например, цветную полосу);
- сохранение истории расчётов в локальную БД (SQLite).

Написать тест на JUnit 5 для проверки корректности вычисления ИМТ.

Задание 6. Создать Spring MVC-приложение «Библиотека» с конфигурацией на аннотациях.

Реализовать:

- сущности: Книга, Автор, Читатель;
- CRUD-операции для книг (добавление, редактирование, удаление, просмотр списка);
- поиск книг по автору и названию;
- использование Thymeleaf для шаблонов.

Написать интеграционный тест на JUnit 5 с использованием @SpringBootTest.

Примерные задания проектной работы (семестр 4)

Задание 1. Информационная система «Складской учёт».

Реализовать систему с веб-интерфейсом для учёта товаров на складе.

Функционал:

- добавление/удаление/редактирование товара;
- поиск по артикулу, названию, категории;
- фильтр по дате поступления и категории товара;
- гистограмма количества поступлений по дням;
- счётчик товаров в таблице;
- подсчёт количества товаров, поступивших за один день.

Параметры объекта «Товар»: ID, название, категория, количество, дата поступления, поставщик, цена.

Система авторизации: роли «Менеджер склада» (полный доступ) и «Кладовщик» (только просмотр и поиск).

Раздел «Новости склада»: административная панель и главная страница

блога с записями (заголовок, дата, автор, текст).

Задание 2. Информационная система «Бронирование отелей».

Создать веб-систему для управления бронированиями. Функционал:

- добавление/удаление/редактирование бронирования;
- поиск по гостю, отелю, датам заезда/выезда;
- фильтр и сортировка по дате заезда;
- гистограмма загруженности отелей по дням;
- счётчик активных бронирований;
- подсчёт количества заездов за день.

Параметры объекта «Бронирование»: ID, гость, отель, дата заезда, дата выезда, номер, статус.

Система ролей: «Администратор» (все функции), «Менеджер» (управление бронированиями), «Гость» (просмотр своих бронирований).

Раздел «Акции отелей»: блог с анонсами специальных предложений.

Задание 3. Информационная система «Управление библиотекой».

Разработать систему учёта книг и читателей. Функционал:

- добавление/удаление/редактирование книг и читателей;
- поиск книг по автору, названию, году издания;
- фильтр по дате выдачи/возврата;
- гистограмма выдачи книг по неделям;
- счётчик книг в каталоге;
- подсчёт количества выданных книг за день.

Параметры объекта «Книга»: ID, название, автор, год издания, статус (на руках/в наличии), читатель (если выдана), дата выдачи, дата возврата.

Роли: «Библиотекарь» (полный доступ), «Читатель» (поиск и просмотр доступных книг).

Раздел «Литературные новости»: блог с обзорами и анонсами мероприятий.

Задание 4. Информационная система «Автосервис».

Создать систему учёта заявок на ремонт. Функционал:

- добавление/удаление/редактирование заявки;
- поиск по клиенту, автомобилю, статусу заявки;
- фильтр по дате приёма автомобиля;
- гистограмма количества принятых автомобилей по дням;
- счётчик активных заявок;
- подсчёт количества принятых автомобилей за день.

Параметры объекта «Заявка»: ID, клиент, автомобиль, дата приёма, предполагаемая дата выдачи, статус, мастер, описание проблемы.

Роли: «Администратор сервиса» (все функции), «Мастер» (работа со своими заявками), «Клиент» (просмотр статуса своих заявок).

Раздел «Советы по эксплуатации»: блог с рекомендациями по уходу за автомобилем.

Задание 5. Информационная система «Фитнес-клуб».

Разработать систему управления абонементом и посещениями.

Функционал:

- добавление/удаление/редактирование абонемента;
- поиск по клиенту, типу абонемента, дате активации;
- фильтр по дате посещения;
- гистограмма посещаемости по дням недели;
- счётчик активных абонементов;
- подсчёт количества посещений за день.

Параметры объекта «Абонемент»: ID, клиент, тип, дата активации, дата окончания, количество посещений, тренер (опционально).

Роли: «Администратор клуба» (полный доступ), «Тренер» (просмотр записей своих клиентов), «Клиент» (просмотр своего абонемента и расписания).

Раздел «Здоровье и фитнес»: блог с советами по тренировкам и питанию.

Задание 6. Информационная система «Служба доставки еды».

Создать систему управления заказами. Функционал:

- добавление/удаление/редактирование заказа;
- поиск по клиенту, статусу заказа, времени доставки;
- фильтр по дате заказа;
- гистограмма количества заказов по часам;
- счётчик текущих заказов;
- подсчёт количества доставленных заказов за день.

Параметры объекта «Заказ»: ID, клиент, состав заказа, адрес доставки, дата и время заказа, статус, курьер (если назначен).

Роли: «Менеджер доставки» (все функции), «Курьер» (просмотр назначенных заказов), «Клиент» (оформление и просмотр своих заказов).

Раздел «Рецепты недели»: блог с кулинарными рецептами и акциями.

Задание 7. Информационная система «Туристическое агентство».

Создать систему управления турами. Функционал:

- добавление/удаление/редактирование тура;
- поиск по направлению, датам, цене;
- фильтр по дате начала тура;
- гистограмма бронирований по месяцам;
- счётчик доступных туров;
- подсчёт количества бронирований за день.

Параметры объекта «Тур»: ID, направление, даты поездки, стоимость, количество мест, статус (доступен/забронирован), менеджер.

Роли: «Директор агентства» (все функции), «Менеджер по туризму» (управление турами), «Клиент» (поиск и бронирование туров).

Раздел «Путешествуй с нами»: блог с рассказами о странах и советами туристам.

Примерные задания курсового проекта (семестр 5)

Реализовать проект в виде информационной системы с веб-интерфейсом. Составить техническую документацию на данную систему с использованием JavaDoc. Создать UNIT-тесты данной системы

Примерная тематика курсового проекта

1. Информационно-справочная система туристического агентства.

Цель: автоматизация работы туристического агентства, упрощение поиска и бронирования туров.

Основные функции:

- каталог туров с фильтрацией по стране, цене, длительности, типу отдыха;
- поиск горящих путёвок;
- бронирование туров с указанием количества туристов;
- учёт клиентов и истории бронирований;
- формирование отчётов по продажам;
- раздел «Новости туризма» с анонсами акций и новых направлений.

Ключевые сущности: тур, клиент, бронирование, страна, отель.

2. Информационно-справочная система железнодорожного вокзала.

Цель: организация оперативного доступа к информации о расписании поездов, билетах и услугах вокзала.

Основные функции:

- просмотр расписания поездов (прибытие/отправление);
- поиск поездов по направлению, дате, времени;
- информация о наличии билетов и их стоимости;
- сведения о платформах и путях;
- справочная информация о вокзальных услугах (камеры хранения, залы ожидания и т. д.);
- интерактивная карта вокзала.

Ключевые сущности: поезд, маршрут, вагон, билет, платформа.

3. Информационно-справочная система аэропорта.

Цель: обеспечение пассажиров и сотрудников актуальной информацией о рейсах и инфраструктуре аэропорта.

Основные функции:

- расписание вылетов и прилётов (в т. ч. с учётом задержек);
- статус рейса (регистрация, посадка, вылет);
- информация о гейтах и терминалах;
- поиск рейсов по номеру, направлению, авиакомпании;
- схема аэропорта с указанием магазинов, кафе, зон отдыха;
- уведомления о статусе рейса на email/SMS.

Ключевые сущности: рейс, авиакомпания, терминал, гейт, пассажир.

4. Информационно-справочная система ресторана.

Цель: оптимизация работы ресторана, повышение качества обслуживания клиентов.

Основные функции:

- электронное меню с категориями блюд (закуски, супы, десерты и т. д.);
- фильтрация блюд по диетическим характеристикам (вегетарианские, без глютена и пр.);
- онлайн-бронирование столиков;
- расчёт предварительной стоимости заказа;
- система отзывов и рейтингов блюд;
- учёт предпочтений постоянных клиентов.

Ключевые сущности: блюдо, категория, заказ, столик, клиент.

5. Информационно-справочная система библиотеки.

Цель: автоматизация учёта книг и обслуживания читателей.

Основные функции:

- поиск книг по автору, названию, году издания, тематике;
- учёт выдачи и возврата книг;
- резервирование книг читателями;
- статистика по популярности авторов и жанров;
- уведомления о приближающемся сроке возврата;
- электронный каталог с аннотациями и рейтингами.

Ключевые сущности: книга, читатель, выдача, автор, жанр.

6. Информационно-справочная система магазина цифровой техники.

Цель: упрощение выбора и покупки электроники для клиентов, оптимизация учёта товаров.

Основные функции:

- каталог товаров с фильтрацией по бренду, цене, характеристикам;
- сравнение моделей по ключевым параметрам;
- наличие товаров в разных филиалах магазина;
- онлайн-бронирование с самовывозом;
- отзывы и рейтинги товаров;
- персональные рекомендации на основе истории покупок.

Ключевые сущности: товар, категория, бренд, филиал, заказ.

7. Информационно-справочная система обработки и доставки заказов пиццерии.

Цель: автоматизация приёма, обработки и отслеживания заказов на доставку пиццы.

Основные функции:

- онлайн-меню с фотографиями и составом блюд;
- добавление товаров в корзину, расчёт стоимости;
- оформление заказа с выбором способа доставки (курьер/самовывоз);
- отслеживание статуса заказа (принят, готов, в пути, доставлен);
- история заказов с возможностью повторного оформления;
- интеграция с картами для расчёта времени доставки.

Ключевые сущности: пицца, ингредиент, заказ, курьер, статус заказа.

8. Информационно-справочная система кинотеатра.

Цель: автоматизация продажи билетов и управления репертуаром кинотеатра.

Основные функции:

- расписание сеансов с указанием зала и времени;
- выбор мест в зале с визуализацией свободных кресел;
- покупка и бронирование билетов онлайн;
- фильтры по жанру, возрасту, рейтингу фильма;
- акции и скидки для постоянных клиентов;
- анонсы предстоящих премьер.

Ключевые сущности: фильм, сеанс, зал, билет, место.

9. Информационно-справочная система театра.

Цель: упрощение покупки билетов и информирования зрителей о репертуаре театра.

Основные функции:

- афиша спектаклей с описанием и составом актёров;
- схема зрительного зала с выбором мест;
- бронирование и покупка билетов;
- информация о творческих коллективах и артистах;
- история посещений и персональные рекомендации;
- виртуальные экскурсии по театру.

Ключевые сущности: спектакль, актёр, роль, билет, ряд, место.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры.

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

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

Примерные вопросы для подготовки к зачету (семестр 4)

1. Платформа Java: Java Development Kit (JDK): состав, назначение. Кроссплатформенность языка Java.
2. Java virtual machine (JVM), JIT-компилятор – определение, свойства, функции. Принципы работы сборщика мусора.
3. Системы сборки проектов. Фреймворк Apache Maven: определение, структура, Maven Coordinates, POM-файл.
4. Типы данных Java: простые и ссылочные. Простые (примитивные) типы данных.
5. Переменные: статические и нестатические. Местные переменные, область видимости переменных. Объявление и инициализация переменных. Константы. Спецификаторы доступа.
6. Комментарии: виды, особенности применения.
7. Операции языка Java – арифметические, отношения и логические, преобразования числовых типов.
8. Символьные строки, методы работы со строками Java.
9. Классы StringBuffer и StringBuilder.
10. Массивы Java: объявление, инициализация. Основные методы класса Arrays. Доступ к элементам массивов, итерация массивов. Двумерные массивы.
11. Управляющие конструкции Java: ветвление, циклы. Цикл foreach.
12. Перечислить и дать описание основных принципов объектно-ориентированного программирования (ООП). Достоинства и недостатки ООП.

13. Определение класса. Объявление класса. Спецификаторы доступа. Отношения между классами Java (наследование, зависимость, агрегирование). Статические члены класса. Переменные класса.
14. Объект класса. Создание объекта. Конструктор класса, конструктор по умолчанию. Ключевое слово `this`. Перегрузка конструкторов. Доступ к переменным экземпляра.
15. Методы, объявление, имя. Статические методы. Доступ к методам. Спецификаторы доступа.
16. Пакеты Java. Импорт пакетов и классов. Статический импорт.
17. Вложенные и внутренние классы Java. Статические и нестатические внутренние классы.
18. Наследование. Подклассы и суперклассы. Доступ к членам класса. Конструкторы при наследовании, ключевое слово `super`.
19. Иерархия наследования Java. Преобразование типов при наследовании. Ключевое слово `instanceof`.
20. Полиморфизм в Java. Перегрузка и переопределение методов.
21. Абстрактные методы и классы Java.
22. Интерфейсы Java: определение интерфейса, реализация интерфейса. Преимущества применения интерфейсов. Переменные интерфейсов. Наследование интерфейсов. Методы по умолчанию. Статические методы интерфейсов.
23. Исключения (exception) Java. Синтаксис объявления исключений. Классификация исключений. Основные классы для работы с исключениями. Исключения при наследовании.
24. Коллекции: Java collections framework. Классификация интерфейсов коллекций. Интерфейс `Collection`.
25. Списки. Интерфейс `List`. Основные классы, реализующие интерфейс `List`. `ArrayList`, особенности, методы. `Comparator`.
26. Интерфейс `Set`. Основные реализации. `HashSet`. `TreeSet`.

27. Очереди в Java, интерфейс Queue, PriorityQueue. Структура, основные методы.

28. Байтовые потоки InputStream и OutputStream. Консольный ввод и вывод Java. Символьные потоки данных. Абстрактные классы Writer, Reader.

29. Чтение и запись файлов. Классы FileInputStream, FileOutputStream. Файловый ввод-вывод с использованием символьных потоков. Классы FileReader и FileWriter.

30. Многопоточное программирование: общие принципы.

31. Класс Thread и интерфейс Runnable: создание потоков, приоритеты потоков.

32. Приостановка и прерывание потоков. Определение момента завершения потока.

33. Синхронизация потоков.

34. Архитектура JDBC (Java DataBase Connectivity). Двух и трехуровневые модели доступа к базе данных. Преимущества и недостатки JDBC.

Примерные вопросы для подготовки к экзамену (семестр 5)

1. Пакеты Java. Импорт пакетов и классов. Статический импорт.

2. Вложенные и внутренние классы Java. Статические и нестатические внутренние классы.

3. Наследование. Подклассы и суперклассы. Доступ к членам класса. Конструкторы при наследовании, ключевое слово super.

4. Иерархия наследования Java. Преобразование типов при наследовании. Ключевое слово instanceof.

5. Полиморфизм в Java. Перегрузка и переопределение методов.

6. Абстрактные методы и классы Java.

7. Интерфейсы Java: определение интерфейса, реализация интерфейса. Преимущества применения интерфейсов. Переменные интерфейсов.

Наследование интерфейсов. Методы по умолчанию. Статические методы интерфейсов.

8. Исключения (exception) Java. Синтаксис объявления исключений. Классификация исключений. Основные классы для работы с исключениями. Исключения при наследовании.

9. Платформа JavaFX, особенности, компоненты.
10. Шаблон MVC (Model-View-Controller) в JavaFX.
11. Классы Stage и Scene в JavaFX.
12. Узлы и графы сцены в JavaFX.
13. Класс Application и жизненный цикл приложения JavaFX.
14. Инструмент визуальной компоновки JavaFX Scene Builder.
15. Компоненты управления Label, TextField в JavaFX.
16. Компонент управления Button в JavaFX.
17. Основные фреймворки и задачи, решаемые Spring.
18. Spring Inversion of Control (IoC) контейнер Spring.
19. Dependency Injection (DI) в Spring.
20. Жизненный цикл объекта Bean Spring.
21. Конфигурация ApplicationContext с помощью xml в Spring.
22. Область видимости Bean в Spring.
23. Фабричные или factory-методы в Spring.
24. Конфигурация ApplicationContext с помощью аннотаций в Spring.
25. Связывание в Spring, аннотация @Autowired.
26. Архитектурный стиль REST.
27. Spring Web-MVC, основная схема и логика работы.
28. Класс DispatcherServlet, его функции.
29. Маппинг в Spring.
30. Интерфейсы HttpServletRequest и HttpServletResponse.
31. Архитектурный стиль CRUD, его соответствие REST и HTTP.
32. Шаблон Data Access Object (DAO).

33. Основные понятия Объектно-реляционного отображения (ORM - Object-Relational Mapping).

34. Спецификация Java Persistence API (JPA).

35. Архитектура ORM Java Persistence API (JPA).

36. Основные аннотации Java Persistence API (JPA).

37. Библиотека Hibernate, основные аннотации.

38. Объявление сущности и таблицы в Hibernate.

39. Интерфейс Session в Hibernate.

40. Ассоциация сущностей в Hibernate.

41. Spring Boot: определение, характеристики, преимущества.

42. Spring Initializr, особенности и преимущества применения.

43. Структура фреймворка JUnit.

44. JUnit аннотации @Test, @DisplayName.

45. JUnit аннотации @BeforeEach, @AfterEach.

46. Тестовые классы и методы JUnit.

47. Утверждения JUnit. Класс Assert.

48. Тестирование исключений JUnit.

49. Генератор документирования Javadoc. Виды комментариев.

50. Дескрипторы Javadoc.

Примеры заданий для подготовки к экзамену (семестр 5)

Задание 1. Разработать конвертер температур (Цельсий ↔ Фаренгейт ↔ Кельвин) с графическим интерфейсом на JavaFX. Реализовать:

- поля ввода/вывода для каждой шкалы;
- кнопки конвертации между всеми парами шкал;
- валидацию ввода (только числа);
- обработку ошибок (например, температура ниже абсолютного нуля).

Выполнить документирование на основе Javadoc. Написать тест к этому приложению на основе JUnit 5, проверяющий корректность вычислений.

Задание 2. Создать приложение «Список дел» (To-Do List) на JavaFX:

- поле для ввода новой задачи;
- кнопка «Добавить»;
- список задач с чекбоксами для отметки выполнения;
- кнопка «Удалить выполненные»;
- сохранение списка в файл при закрытии приложения и загрузка

при запуске.

Выполнить документирование на основе Javadoc.

Задание 3. Разработать калькулятор ИМТ (индекс массы тела) с графическим интерфейсом JavaFX:

- ввод роста (в см) и веса (в кг);
- вычисление ИМТ и вывод категории (недостаток веса, норма, избыток, ожирение);
- визуальную индикацию результата (например, цветную полосу);
- сохранение истории расчётов в локальную БД (SQLite).

Написать тест на JUnit 5 для проверки корректности вычисления ИМТ.

Задание 4. Написать программу, которая анализирует текст, введённый пользователем в поле JavaFX:

- подсчитывает количество слов, предложений, символов;
- находит самые частые слова (топ-5);
- выводит статистику в табличном виде.

Выполнить документирование на основе Javadoc.

Задание 5. Разработать приложение «Календарь событий» на JavaFX:

- выбор даты из календаря;
- поле ввода события;
- кнопка «Добавить событие»;
- отображение всех событий на выбранную дату;
- экспорт событий за месяц в CSV-файл.

Написать тест на JUnit 5 для проверки добавления и поиска событий.

Задание 6. Создать Spring MVC-приложение «Библиотека» с конфигурацией на аннотациях:

- сущности: Книга, Автор, Читатель;
- CRUD-операции для книг (добавление, редактирование, удаление, просмотр списка);
- поиск книг по автору и названию;
- использование Thymeleaf для шаблонов.

Написать интеграционный тест на JUnit 5 с использованием @SpringBootTest.

Задание 7. Разработать Spring Boot-приложение для управления заказами в интернет-магазине:

- REST API для работы с заказами (GET, POST, PUT, DELETE);
- БД: PostgreSQL или H2 (в памяти);
- валидация входящих данных (Bean Validation);
- логирование операций.

Создать тесты для REST-контроллеров с использованием MockMvc.

Задание 8. Реализовать форму регистрации и авторизации пользователя на Spring Boot Security:

- регистрация с подтверждением по email;
- вход с ролями USER и ADMIN;
- ограничение доступа к админ-разделу;
- хранение паролей с хешированием (BCrypt).

Выполнить документирование API с помощью Swagger.

Задание 9. Создать микросервис «Погода» на Spring Boot:

- получение данных о погоде из внешнего API (например, OpenWeatherMap);
- кэширование ответов на 1 час;
- эндпоинт для получения прогноза по городу;
- обработка ошибок (город не найден, проблемы с сетью).

Написать тесты для сервиса с мокированием внешнего API.

Задание 10. Разработать MVC-приложение «Опрос» на Spring Boot + Thymeleaf:

- создание опросов с вопросами и вариантами ответов;
- прохождение опроса анонимными пользователями;
- просмотр статистики по ответам (диаграммы, проценты);
- экспорт результатов в PDF.

Задание 11. Реализовать визуализатор сортировки на JavaFX:

- выбор алгоритма (пузырьковая, быстрая, слиянием);
- генерация случайного массива чисел;
- визуализация процесса сортировки (столбцы разной высоты);
- отображение времени выполнения.

Выполнить документирование на основе Javadoc.

Задание 12. Написать программу для построения графа и поиска кратчайшего пути (алгоритм Дейкстры):

- добавление вершин и рёбер через GUI;
- визуализация графа (круги и линии);
- выбор начальной и конечной вершины;
- подсветка кратчайшего пути;
- вывод длины пути.

Написать тест для алгоритма Дейкстры на JUnit 5.

Задание 13. Разработать приложение для работы с бинарным деревом поиска на JavaFX:

- вставка, удаление и поиск узлов;
- визуализация дерева (узлы и связи);
- обход в ширину и глубину с отображением порядка обхода;
- балансировка дерева (AVL).

Выполнить документирование на основе Javadoc.

Задание 14. Создать генератор лабиринтов на JavaFX с алгоритмом поиска пути:

- генерация лабиринта (алгоритм Эллера или Краскала);

- визуализация (чёрные стены, белые проходы);
- поиск пути от старта к финишу (BFS или A*);
- анимация движения «персонажа» по найденному пути.

Задание 15. Реализовать программу для решения sudoku с графическим интерфейсом:

- поле 9×9 с возможностью ввода цифр;
- проверка корректности ввода;
- кнопка «Решить» (алгоритм обратного отслеживания);
- подсветка ошибок.

Написать тесты для алгоритма решения sudoku.

Задание 16. Написать программу, которая вычисляет интеграл функции методом прямоугольников или трапеций:

- ввод функции (например, $x^2 + 2x + 1$) в текстовом поле;
- указание пределов интегрирования и количества разбиений;
- графическое отображение функции и разбиений на графике;
- вывод результата.

Выполнить документирование на основе Javadoc.

Задание 17. Разработать приложение для анализа CSV-файлов на JavaFX:

- загрузка файла через диалог выбора;
- отображение данных в таблице;
- фильтрация и сортировка по столбцам;
- построение графиков (гистограмма, линейный график) по

числовым данным.

Написать тест для парсера CSV.

Задание 18. Создать конвертер валют с графическим интерфейсом:

- выбор валют из выпадающих списков;
- ввод суммы;
- получение курсов из внешнего API (например, ЦБ РФ или Open

Exchange Rates);

- кэширование курсов на 30 минут;
- история конвертаций.

Выполнить документирование на основе Javadoc.

Задание 19. Написать программу, которая решает систему линейных уравнений методом Гаусса:

- ввод матрицы коэффициентов и вектора свободных членов;
- пошаговое отображение преобразований матрицы;
- вывод корней или сообщение о несовместности системы;
- проверка решения подстановкой.

Написать тест для метода Гаусса на JUnit 5.

Задание 20. Реализовать приложение для шифрования текста (шифр Цезаря или RSA):

- ввод текста и ключа;
- выбор режима (шифрование/дешифрование);
- вывод результата;
- сохранение зашифрованного текста в файл.

Выполнить документирование на основе Javadoc.

Пример экзаменационного билета

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

Кафедра: **Корпоративные инфокоммуникационные системы**
Дисциплина: **Современные технологии программирования**
Филиал: **Владикавказский**; Форма обучения: **Очная**
Семестр: 5 Направление: **09.03.04 Программная инженерия**
Профиль: **Технологии разработки программного обеспечения**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. **Теория.** Виды контейнеров в Swing. (15 баллов)
2. **Теория.** Абстрактные методы и классы Java. (15 баллов)
3. **Задание.** Спроектировать и реализовать веб-приложение калькулятора для выполнения базовых арифметических операций, используя Spring Boot (MVC) и шаблон Thymeleaf для отображения интерфейса. (30 баллов)

Подготовил: _____ М.А Ковалева

На основе перечня теоретических вопросов и заданий, утвержденного на заседании кафедры «Корпоративные инфокоммуникационные системы» протокол № ____ от _____.2026 г.

Утверждаю:
Заведующий кафедрой _____ М.А Ковалева
Дата _____.2026г.

Примеры оценочных средств для проверки индикаторов достижения компетенций, формируемых дисциплиной

Код и наименование компетенции	Наименование индикатора достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ОПК-4. Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	1. Демонстрирует знания основных стандартов ведения технической документации, как отечественных, так и зарубежных.	Знать: основные стандарты ведения технической, как отечественной, так и зарубежной технической документации. Уметь: составлять и вести, как отечественную, так и зарубежную техническую документацию.	Вопросы: 1. Общая информация о языке программирования. 2. Базовые принципы объектно-ориентированного программирования: наследование, инкапсуляция, полиморфизм. 3. Логирование. Документирование Задание Разработать приложение для работы с бинарным деревом поиска: вставка, удаление и поиск узлов; визуализация дерева (узлы и связи); обход в ширину и глубину с отображением порядка обхода; балансировка дерева (AVL). Выполнить документирование на основе Javadoc..
	2. Создает комплект программной и проектной документации к разрабатываемой информационной системе.	Знать: методы создания комплекта программной и проектной документации к разрабатываемой информационной системе. Уметь: создавать комплект программной и проектной документации к разрабатываемой информационной системе.	Вопросы: 1. Интерфейсы Java: определение интерфейса, реализация интерфейса. 2. Класс Application и жизненный цикл приложения JavaFX. Задание Написать программу, которая вычисляет интеграл функции методом прямоугольников или трапеций: ввод функции $x^2 + 2x + 1$ в текстовом поле; указание пределов интегрирования и количества разбиений; графическое отображение функции и разбиений на графике; вывод

			результата. Выполнить документирование на основе Javadoc.
	3. Критически анализирует программную и проектную документацию, строит на ее основе логические выводы.	Знать: методы критического анализа программной и проектной документации. Уметь: критически анализировать программную и проектную документацию; строить на основе программной и проектной документации логические выводы.	Вопросы: 1. Основные понятия Объектно-реляционного отображения (ORM - Object-Relational Mapping). 2. Генератор документирования Javadoc. Виды комментариев 3. Тестовые классы и методы Задание Реализовать приложение для шифрования текста (шифр Цезаря или RSA): ввод текста и ключа; выбор режима (шифрование/дешифрование); вывод результата; сохранение зашифрованного текста в файл. Выполнить документирование на основе Javadoc.
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	1. Демонстрирует знания основных программных продуктов, используемых для решения задач профессиональной деятельности, в том числе, отечественного производства.	Знать: современные интегрированные среды разработки, используемые для решения задач профессиональной деятельности. Уметь: применять современные интегрированные среды при разработке и реализации программных продуктов.	Вопросы: 1. Платформа JavaFX, особенности, компоненты 2. Основные фреймворки и задачи 3. Архитектурный стиль 4. Тестирование исключений Задание Реализовать форму регистрации и авторизации пользователя: регистрация с подтверждением по email; вход с ролями USER и ADMIN; ограничение доступа к админ разделу; хранение паролей с хешированием
	2. Применяет готовые инструментальные средства для решения задач профессиональной деятельности, проводит	Знать: методы инструментальной разработки и возможности их применения при решении профессиональных задач; методы оценки существующих инструментальных	Вопросы: 1. Шаблон MVC 2. Основные фреймворки и задачи 3. Системы автоматической сборки пакетов. Контейнеризация Задание Создать микросервис

	квалифицированную их оценку и обосновывает свой выбор.	средств. Уметь: применять методы инструментальной разработки и оценки существующих инструментальных средств, а также обосновывать их при реализации программных продуктов, в том числе и информационных систем.	«Погода» на Spring Boot: получение данных о погоде из внешнего API (например, OpenWeatherMap); кэширование ответов на 1 час; эндпоинт для получения прогноза по городу; обработка ошибок (город не найден, проблемы с сетью).
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию и тестированию программных продуктов	1. Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования.	Знать: методы разработки и алгоритмы решения простых информационных задач Уметь: разрабатывать и применять алгоритмы решения простых информационных задач	Вопросы: 1. Основные языковые конструкции. 2. Базовые концепции. Структуры данных. 3. Функции, интерфейсы, объектно-ориентированное программирование. Задание Реализовать визуализатор сортировки на JavaFX: • выбор алгоритма (пузырьковая, быстрая, слиянием); • генерация случайного массива чисел; • визуализация процесса сортировки (столбцы разной высоты); отображение времени выполнения.
	2. Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ.	Знать: методы анализа алгоритмов в части производительности и оптимальности; методы создания рекомендаций для оптимизации алгоритмов программ. Уметь: анализировать алгоритмы в части производительности, оптимальности, а также вырабатывать рекомендации для оптимизации алгоритмов программ.	Вопросы: 1. Поточковая организация системы ввода-вывода. Многопоточность. 2. Совместимость на уровне байт-кода. Реализация интерфейсов. Задание Написать программу для построения графа и поиска кратчайшего пути (алгоритм Дейкстры): добавление вершин и рёбер через GUI; визуализация графа (круги и линии); выбор начальной и

			конечной вершины; подсветка кратчайшего пути; вывод длины пути. Написать тест для алгоритма Дейкстры на JUnit 5.
	3. Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев.	Знать: методы черного и белого ящика, составляет набор тестовых случаев, а также методы ручного и автоматизированного тестирования, в том числе и создания UNIT-тестов. Уметь: проводить ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев.	Вопросы: 1. Обработка результатов выполнения запроса 2. Оркестратор, как связующее звено в микросервисной архитектуре. 3. Рутинг, маршрутизация, рендеринг. Обработчики запросов. Тесты. Развертывание. Валидаторы. Задание Написать программу, которая решает систему линейных уравнений методом Гаусса: ввод матрицы коэффициентов и вектора свободных членов; пошаговое отображение преобразований матрицы; вывод корней или сообщение о несовместности системы; проверка решения подстановкой. Написать тест для метода Гаусса на JUnit 5.

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

Основная литература

1. Гуськова, О. И. Объектно ориентированное программирование в Java: учебное пособие / О. И. Гуськова. — Москва: МПГУ, 2018. — 240 с.: ил. — ЭБС Университетская библиотека ONLINE. — URL: <https://biblioclub.ru/index.php?page=book&id=500355>. — Текст: электронный.
2. Васюткина, И. А. Разработка серверной части web-приложений на Java: учебное пособие / И. А. Васюткина. — Новосибирск: НГТУ, 2021. — 83 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/216155>. — Текст: электронный.

Дополнительная литература

3. Пруцков, А. В. Язык программирования Java. Введение в курс: операторы и типы данных: учебное пособие / А. В. Пруцков. — Рязань: РГРТУ, 2016. — 72 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/168307>. — Текст: электронный.
4. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения: учебное пособие для вузов / С. А. Чернышев. — Москва: Юрайт, 2023. — 176 с. — (Высшее образование). — ЭБС Юрайт. — URL: <https://urait.ru/bcode/520097>. - Текст: электронный.

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

1. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
2. Электронно-библиотечная система Znanium <http://www.znaniy.com>
3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.biblio-online.ru>
5. Электронная библиотека издательского дома «Гребенников» <https://grebennikov.ru>
6. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com>

10. Методические указания для обучающихся по освоению дисциплины

Методика освоения дисциплины предусматривает подготовку обучающихся к лекциям, семинарам и практическим занятиям, выполнение студентами самостоятельной внеаудиторной работы.

Рекомендации по подготовке к лекционным занятиям.

Для наиболее полного освоения дисциплины студентам необходимо:

- перед каждой лекцией просматривать рабочую программу дисциплины, ее основные вопросы и рекомендуемую литературу. Это позволит сэкономить время на записывание основных вопросов темы;
- перед очередной лекцией просматривать материалы предыдущих, чтобы освоение материала не оставляло пробелов.

Рекомендации по подготовке к семинарам, практическим занятиям.

Студентам следует:

- проработать теоретический материал к занятию по рекомендованным литературным источникам и лекциям;
- использовать при подготовке к занятию нормативно-правовые документы, научные публикации, информационный материал, рекомендуемый преподавателем;
- перед занятиями задать вопросы по невыясненным в ходе самостоятельной подготовки темам или отдельным положениям темы;
- в ходе занятия давать четкие и исчерпывающие ответы на вопросы;
- на занятии демонстрировать понимание обсуждаемых тем и вопросов.

Студентам, пропустившим занятия по различным причинам, необходимо перед очередным занятием отработать пропущенный материал, подготовив его самостоятельно.

Методические рекомендации по выполнению различных форм самостоятельной работы

Студентам при организации самостоятельной работы следует руководствоваться Приказом Финансового университета № 1040/о от 11.05.2021г. «Об утверждении методических рекомендаций по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете».

Самостоятельная работа содержит в себе различные виды и формы

работ. Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По теме учебной дисциплины студентам предлагается перечень заданий для самостоятельной работы.

В ходе изучения дисциплины предусмотрены следующие формы самостоятельной работы:

- подготовка к опросу;
- разбор вопросов, отводимых на самостоятельное освоение,
- решение задач;
- выполнение проектной работы;
- выполнение курсового проекта;
- подготовка к зачету и экзамену.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также должны соответствовать установленным требованиям по оформлению.

Студентам следует:

- руководствоваться графиком самостоятельной работы, определенным РПД;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, разбирать на занятиях и консультациях неясные вопросы;
- прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные фрагменты для их обсуждения на консультации.

Методические рекомендации для обучающихся по выполнению проектной работы

Проектная работа является обязательной формой внеаудиторной самостоятельной работы студентов по дисциплине.

Целью проектной работы является развитие у студентов способности

прогнозировать, проектировать, моделировать, формирование учебно-исследовательских навыков, закрепление умений самостоятельно работать с различными источниками информации; проверка сформированности компетенций.

Проектная работа может выполняться как индивидуально, так и в составе группы. Количество групп и их численный состав определяет преподаватель, ведущий семинарские занятия.

Заказчиками выполнения проекта могут являться представители работодателей. В этом случае проектная работа выполняется исходя из потребностей заказчика.

Выполнение проекта предполагает:

- диагностику ситуации (проблематизация, целеполагание, конкретизация цели, форматирование проекта);
- проектирование (уточнение цели, функций, задач и плана работы; теоретическое моделирование методов и средств решения задач; детальная проработка этапов решения конкретных задач; пошаговое выполнение запланированных проектных действий; систематизация и обобщение полученных результатов, конструирование предполагаемого результата, пошаговое выполнение проектных действий);
- рефлексия (выяснение соответствия полученного результата замыслу; определение качества полученного продукта; перспективы его развития и использования);
- фиксация результатов в виде исполненного проекта.

Проектная работа состоит из нескольких частей. Состав проектной работы и очередность размещения отдельных частей:

- титульный лист;
- основная часть;
- список использованных источников;
- приложения (при наличии).

Титульный лист является первой страницей проектной работы и

заполняется по определенным правилам.

Основная часть выполняется согласно заданию преподавателя или исходя из потребностей заказчика.

В список использованных источников включаются названия законодательных актов, нормативных документов, книг, статей, учебных пособий и т. п., которые, так или иначе, использовались студентом при выполнении работы.

Иллюстративный материал (схемы, диаграммы, рисунки, таблицы и др.) встраивается в текст работы или выносится в Приложения. В Приложения выносятся вспомогательные материалы, которые не содержат основную информацию, либо материалы, которые сложно разместить по тексту работы (большие схемы, таблицы, графические материалы, расчетные справочные данные, образцы первичных документов и т.п.). Непременным условием включения данных материалов в приложение является ссылка на них в тексте работы.

Требования к оформлению проектной работы.

Проектная работа выполняется на компьютере на одной стороне белой бумаги формата А4 (210х297 мм). Размер шрифта -13 или 14, междустрочный интервал – одинарный или полуторный.

Размеры полей: левое - 30 мм, правое - 15 мм, верхнее - 20 мм, нижнее - 20 мм. Отступ первой строки абзаца - 1,25. Нумерация страниц – внизу в центре.

При написании допускаются только общепринятые сокращения (например, тыс. руб.).

Общий объем проектной работы составляет не более 10 страниц, не включая таблицы, графики и т.п. (при наличии), а также приложения (при наличии).

В тексте обязательны ссылки на литературные источники, лучше всего постраничные.

Законченная проектная работа, содержащая все требуемые элементы оформления, вставленная в папку (или файл) и скрепленная по левому краю, сдается на кафедру или непосредственно руководителю проектной работы – преподавателю; ведущему семинарские (практические) занятия по дисциплине. Он осуществляет проверку проектной работы, а также оказывает помощь при подготовке к ее защите.

Проектная работа защищается в назначенные сроки. Защита проектной работы проводится до начала сессии (в крайнем случае, до начала экзамена по соответствующему предмету). При защите студент кратко излагает основные положения работы, последовательность ее выполнения, свои предложения.

При защите проектной работы студент должен свободно ориентироваться в изложенном материале работы; ответить на все замечания преподавателя; уметь отвечать на вопросы преподавателя по проектной работе.

Оценка проектных работ студентов проводится в процессе текущего контроля успеваемости.

Критерии оценки проектной работы

Оценка «отлично» (5-6 баллов) выставляется студенту, если проектная работа отличается творческим (креативным) подходом, собственным оригинальным отношением автора к идее проекта; содержит полную диагностику ситуации, а также теоретическое моделирование методов и детальную проработку этапов решения конкретных задач; в работе сделаны необходимые выводы, намечены перспективы использования проекта, спланированы действия по его продвижению; работа отличается грамотным оформлением в точном соответствии с установленными правилами, с соблюдением логической последовательности изложения материала; студент в работе выдвигает новые идеи и трактовки, демонстрирует способность анализировать материал; на дополнительные вопросы при защите проектной работы даны полные ответы.

Оценка «хорошо» (3-4 балла) выставляется студенту, если проектная работа содержит достаточно полную диагностику ситуации, а также теоретическое моделирование методов и этапов решения конкретных задач; в работе сделаны выводы, намечены перспективы использования проекта; работа оформлена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно или по требованию преподавателя; в работе соблюдена логическая последовательность изложения материала; студент в работе демонстрирует творческие способности и хорошую способность анализировать материал. На дополнительные вопросы при защите проектной работы даны не совсем полные ответы.

Оценка «удовлетворительно» (1-2 балла) выставляется студенту, если проектная работа содержит отдельные элементы моделирования методов и этапов решения конкретных задач; в работе сделаны выводы, намечены перспективы использования проекта; работа выполнена и оформлена правильно, но в ней допущены 1-2 погрешности или одна грубая ошибка; в работе соблюдена логическая последовательность изложения материала; студент в работе демонстрирует удовлетворительную способность анализировать материал; допущены ошибки при ответе на дополнительные вопросы при защите проектной работы.

Оценки «неудовлетворительно» (0 баллов) заслуживает студент, если в работе отсутствуют элементы моделирования; студент в работе не проявил способность анализировать, прогнозировать и проектировать; в работе отсутствует логическая последовательность изложения материала, допущены грубые ошибки, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

При оценивании проектной работы на «неудовлетворительно» работа должна быть переделана (исправлена) в соответствии с полученными замечаниями, сдана на проверку заново и защищена не позднее срока окончания ее приёма и защиты.

Методические указания по выполнению курсового проекта

Студентам при выполнении курсового проекта следует руководствоваться Приказом Финансового университета № 1583/о от 02.06.2021г. «Об утверждении Положения о курсовом проектировании по образовательным программам высшего образования – программам бакалавриата и магистратуры в Финансовом университете».

Курсовой проект– важная форма изучения студентами дисциплины и одна из форм обязательной внеаудиторной самостоятельной учебной и научной работы студентов, осуществляемая под руководством преподавателя.

Целью курсового проекта является расширение, систематизация и закрепление теоретических и практических знаний, полученных студентами, а также выработка у студентов практических навыков самостоятельной работы.

Выполнение курсового проекта ориентировано на решение следующих задач:

- закрепление, углубление теоретических знаний и практическая реализация умений и навыков, полученных в ходе лекционных и практических (семинарских) занятий по дисциплине, по конкретной проблематике, практическая их реализация;
- получение опыта самостоятельного проведения исследования;
- формирование умения работать с источниками информации, Интернет-ресурсами, базами данных, информационно-аналитическими системами, навыков анализа и интерпретации показателей;
- развитие способности формулирования самостоятельных обоснованных выводов и рекомендаций, подготовки информационных и аналитических обзоров;
- развитие и повышение творческого потенциала студентов;
- получение опыта и формирование навыков подготовки, оформления и защиты научных работ для дальнейшего их использования при написании выпускной квалификационной работы.

Процесс подготовки и выполнения курсового проекта включает следующие этапы:

- 1) выбор из предложенного примерного списка темы курсового проекта и закрепление руководителя;
- 2) составление плана работы и согласование его с руководителем;
- 3) подбор литературы и сбор необходимой исходной информации по избранной теме; изучение выбранных источников литературы, а также обработка, систематизация, обобщение и анализ собранного фактического материала и полученной информации;
- 4) выполнение курсового проекта, формулирование выводов и рекомендаций; оформление курсового проекта в соответствии с требованиями;
- 5) подготовка к защите и защита курсового проекта.

Курсовой проект состоит из нескольких частей. Состав курсового проекта и очередность размещения отдельных частей в ней следующий:

- титульный лист (одна страница);
- содержание (одна страница);
- введение (2 страницы);
- основная часть, включающая две главы (20-25 страниц);
- заключение (2-3 страницы);
- список использованных источников (25-30 источников);
- приложения (при необходимости).

В процессе написания курсового проекта студент должен разобраться в теоретических вопросах избранной темы, самостоятельно проанализировать практический материал, разобрать и обосновать практические предложения.

Таким образом, качественно выполненная курсового проекта характеризуется двумя основными ее составляющими: во-первых, раскрытие экономической сущности изучаемой проблемы и изложение собственной позиции по дискуссионным вопросам; во-вторых, глубокий и всесторонний анализ действующей практики, исходя из конкретного фактического

материала. Простое переписывание прочитанного материала; изложение дискуссионных вопросов без формирования собственной позиции; описание текущих инструкций, без аналитического осмысливания практического материала может послужить причиной низкой оценки, выполненного курсового проекта. Т.к. такая работа не отражает умение автора самостоятельно и творчески использовать имеющийся материал и сочетать его с теоретическими знаниями, полученными при изучении данной дисциплины.

При подготовке основной части курсового проекта обязательными являются ссылки на использованные студентом источники (научную, методическую или учебную литературу, научные публикации и т.д.).

Курсовой проект, оформленный в соответствии с требованиями, подписывается студентом и представляется в электронном и бумажном носителях на кафедру не позднее, чем за три недели до установленного срока защиты курсового проекта.

Курсовой проект в обязательном порядке проверяется руководителем курсового проекта в системе «Антиплагиат». Руководитель проверяет курсовой проект и составляет о ней письменный отзыв, и при условии положительной оценки допускает курсовой проект к защите.

Завершающим этапом выполнения студентом курсового проекта является ее защита. Конкретную дату защиты устанавливает научный руководитель. В случае неявки студента в назначенную научным руководителем дату на защиту курсового проекта по уважительной причине, подтвержденной документально, научным руководителем устанавливается еще одна дата защиты курсового проекта в рамках установленного срока защиты курсовых работ.

На защите студент должен хорошо ориентироваться в представленной работе, уметь дать пояснения по использованным источникам информации, примененным методам исследования, полученным результатам, свободно

отвечать на вопросы как теоретического, так и практического характера, относящиеся к теме работы.

Защита курсового проекта позволяет оценить полноту знания студентом исследованной темы, степень самостоятельности ее выполнения, уровень освоения им компетенций.

Оценка курсового проекта определяется преподавателем в процессе собеседования.

Основными критериями при оценивании курсового проекта являются:

1. Своевременное составление плана и библиографии по работе.
2. Актуальность темы исследования.
3. Правильное оформление работы.
4. Соответствие работы методическим рекомендациям.
5. Степень раскрытия темы.
6. Наличие анализа и обобщения по исследованию.
7. Аргументированность заключений по исследованию.
8. Процедура защиты – ответы на вопросы и замечания.

Конкретное распределение баллов отражается в Отзыве на курсовой проект.

Оценка результатов текущего контроля успеваемости и промежуточной аттестации обучающихся осуществляется в соответствии с Балльно-рейтинговой системой Финансового университета (Приказ Финансового университета № 2187/о от 01.10.2024 г. «Об утверждении Положения о проведении текущего контроля успеваемости и промежуточной аттестации студентов, обучающихся по образовательным программам высшего образования в Финансовом университете»).

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

11.1 Комплект лицензионного программного обеспечения

- 1) Антивирусная защита Kaspersky Security для виртуальных и облачных сред;
- 2) Windows, Microsoft Office или Astra Linux, Libre Office.

11.2 Современные профессиональные базы данных, и информационные справочные системы

1. Информационно-правовая система «Гарант»: <https://www.garant.ru>
2. Большая Российская энциклопедия: <https://bigenc.ru/>
3. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>.

11.3 Сертифицированные программные и аппаратные средства защиты информации

Не используются

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

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 45

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол компьютерный – 1 шт.

Стол (студенческий) двухместный – 13 шт.

Стулья – 27 шт.

Доска меловая – 1 шт.

Технические средства обучения:

Компьютер в сборе – 1 шт.

Доска интерактивная – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 47

Специализированная мебель:

Стол компьютерный – 20 шт.

Стол (двухместный) – 7 шт.

Стул – 34 шт.

Шкаф – 1 шт.

Технические средства обучения:

Компьютер в сборе – 20 шт.

Мультимедиа-проектор – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Помещение для самостоятельной работы обучающихся

Кабинет № 55. Читальный зал:

Специализированная мебель:

Стол – 20 шт.

Стул – 40 шт.

Шкаф для книг – 4 шт.

Стеллаж книжный – 13 шт.

Стеллаж выставочный – 4 шт.

Технические средства обучения:

Компьютер в сборе – 6 шт.

Телевизор – 1 шт.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Финансового университета