

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Департамент анализа данных и машинного обучения
Факультета информационных технологий и анализа больших данных**

УТВЕРЖДАЮ

**Проректор по учебной и
методической работе**

_____ **Е.А. Каменева**

28.12.2023 г.

Д.М. Милованов, Е.В. Пальчевский

Современные технологии программирования

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 – Прикладная информатика,

ОП «Инженерия данных»,

ОП «Прикладные информационные системы в экономике и финансах»,

ОП «Цифровые платформы управления предприятиями»

Рекомендовано Ученым советом

*Факультета информационных технологий и анализа больших данных
(протокол № 39 от 20.12.2023 г.)*

Одобрено Советом учебно-научного

*Департамента анализа данных и машинного обучения
(протокол № 10 от 14.12.2023 г.)*

Москва 2023

Содержание

| | | |
|--|--|----|
| 1. | Наименование дисциплины..... | 3 |
| Ошибка! Закладка не определена. | | |
| 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 образовательных программ..... | 4 |
| 4. | Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся..... | 4 |
| 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..... | 16 |
| 6. | Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине..... | 19 |
| 6.1. | Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы..... | 19 |
| 6.2. | Перечень вопросов, заданий, тем для подготовки к текущему контролю..... | 20 |
| 7. | Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине..... | 23 |
| 8. | Перечень основной и дополнительной литературы, необходимой для освоения дисциплины..... | 32 |
| 9. | Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины..... | 33 |
| 10. | Методические указания для обучающихся по освоению дисциплины..... | 34 |
| 11. | Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем..... | 35 |
| 12. | Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине..... | 35 |

1. Наименование дисциплины

«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 |
|-----------------|---|--|---|
| ПКН-1 | Способность применять общенаучные, общетехнические знания, математические методы в сфере ИТ | 1. Демонстрирует знания о современных естественно-научных концепциях, общетехнических подходах, методах математического анализа и моделирования. | Знать современные естественно-научные концепции, возможности аналитических информационных систем, общетехнические подходы и методы математического анализа и моделирования в рамках разработки программного обеспечения. Уметь применять информационные технологии для математического анализа и моделирования при разработке программного обеспечения. |
| | | 2. Применяет знания для теоретического и экспериментального исследования в сфере разработки программного обеспечения. | Знать современные методы теоретического и экспериментального исследования при управлении и разработкой проектов в области программного обеспечения. Уметь применять современные гибкие методы разработки программного обеспечения, включающие задачи планирования, анализа требований, проектирования, программирования, тестирования и документирования; осуществлять выбор программных и инструментальных средств для разработки, создания и отладки программного обеспечения; использовать готовые программные решения; организовывать верификацию, тестирование и проверку стабильности работы программного обеспечения. |

3. Место дисциплины в структуре образовательных программ

Дисциплина «Современные технологии программирования» относится к Общефакультетскому (предпрофильному) циклу дисциплин по направлению подготовки 09.03.03 – Прикладная информатика, ОП «Инженерия данных», ОП «Прикладные информационные системы в экономике и финансах», ОП «Цифровые платформы управления предприятиями».

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

очная форма обучения

| Вид учебной работы по дисциплине | Всего (в з/е и часах) | Семестр 4 (в часах) | Семестр 5 (в часах) |
|--|--------------------------|------------------------|---|
| Общая трудоемкость дисциплины (в том числе курсовая работа/ курсовой проект) | 6/216 | 108 | 108 (курсовая работа/ курсовой проект- 24 часа) |
| <i>Контактная работа – Аудиторные занятия</i> | 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 | Проектная работа | – |
| Вид промежуточной аттестации | Зачет, экзамен | Зачет | Экзамен |

очно-заочная форма обучения, очно-заочная форма обучения (ИОО)

| Вид учебной работы по дисциплине | Всего (в з/е и часах) | Семестр 4 (в часах) | Семестр 5 (в часах) |
|--|--------------------------|------------------------|---|
| Общая трудоемкость дисциплины (в том числе курсовая работа/ курсовой проект) | 6/216 | 108 | 108 (курсовая работа (курсовой проект) - 24 часа) |
| <i>Контактная работа – Аудиторные занятия</i> | 68 | 34 | 34 |
| <i>Лекции</i> | 32 | 16 | 16 |
| <i>Семинары, практические занятия</i> | 36 | 18 | 18 |
| Самостоятельная работа | 148 | 74 | 74 |
| Вид текущего контроля | Проектная работа | Проектная работа | – |
| Вид промежуточной аттестации | Зачет, экзамен | Зачет | Экзамен |

Заочная форма обучения (ИОО)

| Вид учебной работы по дисциплине | Всего (в з/е и часах) | Семестр 5 (в часах) | Семестр 6 (в часах) |
|---|--------------------------|------------------------|---|
| Общая трудоемкость дисциплины (в том числе курсовая работа/ курсовой проект) | 6/216 | 108 | 108 (курсовая работа /курсовой про- ект) - 24 часа) |
| <i>Контактная работа – Аудиторные занятия</i> | <i>30</i> | <i>16</i> | <i>14</i> |
| <i>Лекции</i> | <i>6</i> | <i>4</i> | <i>2</i> |
| <i>Семинары, практические занятия</i> | <i>24</i> | <i>12</i> | <i>12</i> |
| Самостоятельная работа | 186 | 92 | 94 |
| Вид текущего контроля | - | - | - |
| 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. Закрывание 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 ре-сивера. 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. Закрытие 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

очная форма обучения

| № п/ п | Наименование тем (разделов) дисциплины | Трудоёмкость в часах | | | | | Форма текущего контроля успеваемости |
|--------------|---|----------------------|---|-------------|--------------------------------------|--------------------------------|---|
| | | Всего | Контактная работа - Аудиторная работа* | | | Самосто- ятельная работа | |
| | | | Общая, в т.ч.: | Лек- ции | Семинары, практические занятия | | |

| | | | | | | | |
|---|--|----|---|---|---|---|---|
| 1 | Обзор платформы и языка программирования Java | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 2 | Основные языковые конструкции Java. Коллекции. | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 3 | Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ. | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 4 | Потоковая организация системы ввода-вывода. Многопоточность. | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 5 | Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование. | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 6 | GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты. | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 7 | Микросервисная архитектура. Оркестратор. Backend-Frontend. | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 8 | Работа с базами данных. Основы объектно-реляционного отображения. Использование | 16 | 8 | 2 | 6 | 8 | Тестовые задания, устный опрос, проверка практических заданий |

| | | | | | | | |
|----|--|-----|-----|----|----|-----|---|
| | ORM в различных языках программирования. Механизм миграций. | | | | | | |
| 9 | Системы автоматической сборки пакетов. Контейнеризация. Docker. | 14 | 6 | 2 | 4 | 8 | Тестовые задания, устный опрос, проверка практических заданий |
| 10 | Spring Framework. Архитектура, основные сведения | 18 | 8 | 4 | 4 | 10 | Тестовые задания, устный опрос, проверка практических заданий |
| 11 | RESTful API приложения. | 18 | 8 | 4 | 4 | 10 | Тестовые задания, устный опрос, проверка практических заданий |
| 12 | GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения. | 20 | 8 | 4 | 4 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 13 | Логирование. Документирование. | 18 | 6 | 2 | 4 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| | В целом по дисциплине | 216 | 100 | 32 | 68 | 116 | Согласно учебному плану: проектная работа |
| | Итого в % | | 46 | 32 | 68 | 54 | |

очно-заочная форма обучения, очно-заочная форма обучения (ИОО)

| № п/ п | Наименование тем (разделов) дисциплины | Трудоёмкость в часах | | | | | Форма Текущего контроля успеваемости |
|--------------|---|----------------------|---|-------------|--------------------------------------|--------------------------------|--|
| | | Всего | Контактная работа - Аудиторная работа* | | | Самосто- ятельная работа | |
| | | | Общая, в т.ч.: | Лек- ции | Семинары, практические занятия | | |
| 1 | Обзор плат- формы и языка программирова- ния Java | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, про- верка практических заданий |
| 2 | Основные язы- ковые конструк- ции Java. Кол- лекции. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, про- верка практических заданий |

| | | | | | | | |
|---|---|----|---|---|---|----|---|
| 3 | Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 4 | Потоковая организация системы ввода-вывода. Многопоточность. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 5 | Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 6 | GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 7 | Микросервисная архитектура. Оркестратор. Backend-For-Frontend. | 18 | 8 | 4 | 4 | 10 | Тестовые задания, устный опрос, проверка практических заданий |
| 8 | Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных языках программирования. Механизм миграций. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |

| | | | | | | | |
|----|--|-----|----|----|----|-----|---|
| 9 | Системы автоматической сборки пакетов. Контейнеризация. Docker. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 10 | Spring Framework. Архитектура, основные сведения | 16 | 6 | 2 | 4 | 10 | Тестовые задания, устный опрос, проверка практических заданий |
| 11 | RESTful API приложения. | 18 | 8 | 4 | 4 | 10 | Тестовые задания, устный опрос, проверка практических заданий |
| 12 | GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения. | 18 | 8 | 4 | 4 | 10 | Тестовые задания, устный опрос, проверка практических заданий |
| 13 | Логирование. Документирование. | 18 | 6 | 2 | 4 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| | В целом по дисциплине | 216 | 68 | 32 | 36 | 148 | Согласно учебному плану: проектная работа |
| | Итого в % | | 31 | 47 | 53 | 69 | |

Заочная форма обучения (ИОО)

| № п/ п | Наименование тем (разделов) дисциплины | Трудоёмкость в часах | | | | | Форма Текущего контроля успеваемости |
|--------------|--|----------------------|---|-------------|--------------------------------------|--------------------------------|--|
| | | Всего | Контактная работа - Аудиторная работа* | | | Самосто- ятельная работа | |
| | | | Общая, в т.ч.: | Лек- ции | Семинары, практические занятия | | |
| 1 | Обзор плат- формы и языка программирова- ния Java | 18 | 4 | 2 | 2 | 14 | Тестовые задания, устный опрос, про- верка практических заданий |
| 2 | Основные язы- ковые конструк- ции Java. Кол- лекции. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, про- верка практических заданий |
| 3 | Базовые прин- ципы объектно- ориентирован- ного программи- рования. Классы, методы и объекты Java. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, про- верка практических заданий |

| | | | | | | | |
|----|---|----|---|---|---|----|---|
| | Интерфейсы. Исключения. Отладка программ. | | | | | | |
| 4 | Потоковая организация системы ввода-вывода. Многопоточность. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, проверка практических заданий |
| 5 | Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, проверка практических заданий |
| 6 | GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты. | 18 | 4 | 2 | 2 | 14 | Тестовые задания, устный опрос, проверка практических заданий |
| 7 | Микросервисная архитектура. Оркестратор. Backend-Frontend. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, проверка практических заданий |
| 8 | Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных языках программирования. Механизм миграций. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, проверка практических заданий |
| 9 | Системы автоматической сборки пакетов. Контейнеризация. Docker. | 16 | 2 | - | 2 | 14 | Тестовые задания, устный опрос, проверка практических заданий |
| 10 | Spring Framework. Архитектура, основные сведения | 18 | 2 | - | 2 | 16 | Тестовые задания, устный опрос, проверка практических заданий |

| | | | | | | | |
|----|--|-----|----|----|----|-----|---|
| 11 | RESTful API приложения. | 18 | 2 | - | 2 | 16 | Тестовые задания, устный опрос, проверка практических заданий |
| 12 | GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения. | 16 | 4 | 2 | 2 | 12 | Тестовые задания, устный опрос, проверка практических заданий |
| 13 | Логирование. Документирование. | 16 | - | - | - | 16 | Тестовые задания, устный опрос, проверка практических заданий |
| | В целом по дисциплине | 216 | 30 | 6 | 24 | 186 | Согласно учебному плану: - |
| | Итого в % | | 14 | 20 | 80 | 86 | |

* объем контактной работы в очно-заочной/заочной формах обучения и индивидуальных учебных планах определяется соответствующими учебными планами. Темы, реализуемые в виде контактной работы, определяются преподавателем самостоятельно, исходя из уровня их сложности

5.3. Содержание семинаров, практических занятий

| Наименование тем (разделов) дисциплины | Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника) | Формы проведения занятий |
|--|---|---|
| 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. <i>Рекомендуемые источники: 8.1, 8.3</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
| Основные языковые конструкции Java. Коллекции. | Консольный ввод-вывод данных, команды ввода-вывода. Переменные и основные типы 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 | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |

| | | |
|---|---|--|
| | <p>ArrayList. TreeMap. Применение очередей. Алгоритмы: сортировка, двоичный поиск.</p> <p><i>Рекомендуемые источники: 8.1, 8.2</i></p> | |
| <p>Базовые принципы объектно-ориентированного программирования. Классы, методы и объекты Java. Интерфейсы. Исключения. Отладка программ.</p> | <p>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.</p> <p>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</p> <p>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.</p> <p><i>Рекомендуемые источники: 8.1, 8.2</i></p> | <p>Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа</p> |
| <p>Потоковая организация системы ввода-вывода. Многопоточность.</p> | <p>Консольный ввод, запись консольного вывода. Классы ByteArrayInputStream и ByteArrayOutputStream. Классы FileInputStream и FileOutputStream. StringBufferInputStream. Классы FilterInputStreem и FilterOutputStream. BufferedInputStream и BufferedOutputStream. DataInputStream и DataOutputStream. Класс Thread и интерфейс Runnable.</p> <p>Жизненный цикл потока. Фоновые задачи. Управление потоками и приоритеты потоков. Синхронизация. Приостановка, возобновление и остановка потока.</p> <p><i>Рекомендуемые источники: 8.1, 8.2</i></p> | <p>Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа</p> |
| <p>Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование.</p> | <p>Переменные и константы. Типизация. Перевод типов данных. Условная конструкция. Построение и использование функций. Интерфейсы. Структуры и методы. Цикл. Условная конструкция для управления ошибками.</p> <p><i>Рекомендуемые источники: 8.2, 8.3</i></p> | <p>Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа</p> |
| <p>GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты.</p> | <p>Горутин. Канал. Конкурентность. Параллелизм. Канал взаимодействия горутинов.</p> <p><i>Рекомендуемые источники: 8.2, 8.3</i></p> | <p>Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа</p> |

| | | |
|---|--|---|
| Микросервисная архитектура. Оркестратор. Backend-For-Frontend. | Построение микросервисов. Создание оркестратора. Создание нескольких микросервисов и взаимосвязь через оркестратор. <i>Рекомендуемые источники: 8.3</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
| 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. <i>Рекомендуемые источники: 8.2, 8.3</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
| Системы автоматической сборки пакетов. Контейнеризация. Docker. | Сборка проекта при помощи Maven. Основы синтаксиса Maven. Сборка проекта при помощи Gradle. Синтаксис Groovy. Контейнер. Контейнеризация. Docker. <i>Рекомендуемые источники: 8.1-8.3</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
| 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. <i>Рекомендуемые источники: 8.1</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
| 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. <i>Рекомендуемые источники: 8.1</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
| GIN. Архитектура, основные сведения. FastAPI. Архитектура и основные сведения. | Интеграция GIN. Архитектура. ShouldBindUri(). C.JSON. ListenAndServe() FastAPI. Typing Union. @app. @app.get, @app.post(). <i>Рекомендуемые источники: 8.2, 8.3</i> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |

| | | |
|--------------------------------|---|---|
| Логирование. Документирование. | <p>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.</p> <p><i>Рекомендуемые источники: 8.1, 8.2, 8.3</i></p> | Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа |
|--------------------------------|---|---|

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 |
|--|---|--|
| Обзор платформы и языка программирования Java | Установка IDE IntelliJ Idea. Переменные среды JDK. Регистрация github, создание проекта для выполнения семинарских занятий. | Обзор литературы и веб-источников |
| Основные языковые конструкции Java. Коллекции. | Создание блоков кода. Стилизовое оформление программного кода. Ключевые слова. Укороченные логические операции. Многоступенчатые конструкции «if – else – if». Оператор Switch. Цикл без тела. Вложенные циклы. | Обзор литературы и веб-источников |
| 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. | Обзор литературы и веб-источников |
| Потоковая 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. | Обзор литературы и веб-источников |
| Основные языковые конструкции GoLang. Базовые концепции. Структуры данных. Функции, интерфейсы, объектно-ориентированное программирование. | Установка GoLang. Использование Структур. Вызов методов структур. Использование указателей при передаче массивов в функции. | Обзор литературы и веб-источников |

| | | |
|---|---|-----------------------------------|
| GoLang конкурентность и параллелизм. Горутины и каналы. Модули и пакеты. | Создание горутина. Создания канала. Создание модуля. Объединение модулей в пакеты. Параллелизм. | Обзор литературы и веб-источников |
| Микросервисная архитектура. Оркестратор. Backend-For-Frontend. | Построение backend-for-fronted архитектуры. Построение оркестраторной архитектуры. | Обзор литературы и веб-источников |
| Работа с базами данных. Основы объектно-реляционного отображения. Использование ORM в различных языках программирования. Механизм миграций. | Исключения (SQL Exception). Поток в JDBC. Основные этапы работы с базой данных при использовании JDBC. Уровни изоляции транзакций в JDBC. Хранимая процедура. | Обзор литературы и веб-источников |
| Системы автоматической сборки пакетов. Контейнеризация. Docker. | Сборка пакета Maven. Сборка пакета Gradle. Конфигурация Groove. Конфигурация application.yaml. Запуск Docker. | Обзор литературы и веб-источников |
| Spring Framework. Архитектура, основные сведения | Аспектно-ориентированный Spring. Выбор точек сопряжения в описаниях срезов. Объявление аспектов в XML. Аннотирование аспектов. Внедрение аспектов AspectJ. | Обзор литературы и веб-источников |
| RESTful API приложения. | Построение REST API приложения при помощи SPRING. | Обзор литературы и веб-источников |
| 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. | Обзор литературы и веб-источников |
| Логирование. Документирование. | Формирование логирования при помощи соответствующих библиотек. | Обзор литературы и веб-источников |

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

Примерные задания проектной работы

Задание 1.

Реализовать проект в виде информационной системы «Перевозка грузов» с веб-интерфейсом. Необходимый функционал: добавление груза; удаление груза; редактирование груза; поиск груза по различным параметрам; отдельный фильтр (сортировка) по дате поставки груза; гистограмма количества грузов по дням; счетчик грузов в таблице; подсчет количества прихода грузов за один день; система регистрации и авторизации с разграничением прав доступа (какой функци-

онал будет доступен определенной группе пользователей определяет сам разработчик); раздел «Автоблог» (ссылка на него должна быть в верхнем меню), в котором должны быть доступны следующие подразделы: «Административная панель управления блогом», в котором необходимо добавлять/удалять/редактировать записи; «Главная страница блога», на которой выводятся все добавленные через административную панель управления записи (Название записи, дата, кто добавил, текст записи). Параметры объекта «Груз»: ID, Название груза, Содержимое груза, Город отправки груза, Дата отправки груза, Город прибытия груза, Дата прибытия груза.

Задание 2.

Реализовать проект в виде информационной системы «Кинотеатр» с веб-интерфейсом. Необходимый функционал: добавление сеанса; удаление сеанса; редактирование сеанса; поиск сеанса по различным параметрам; отдельный фильтр (сортировка) по дате сеанса; гистограмма количества сеансов; счетчик сеансов в таблице; подсчет количества сеансов за один день; система регистрации и авторизации с разграничением прав доступа (какой функционал будет доступен определенной группе пользователей определяет сам разработчик); раздел «Новости кинотеатра» (ссылка на него должна быть в верхнем меню), в котором должны быть доступны следующие подразделы: «Административная панель управления блогом», в котором необходимо добавлять/удалять/редактировать записи; «Главная страница блога», на которой выводятся все добавленные через административную панель управления записи (Название записи, дата, кто добавил, текст записи). Параметры объекта «Сеанс»: ID, Название фильма, Дата и время сеанса, Количество билетов на сеанс.

Примерная тематика курсового проекта (курсовой работы)

1. Информационно-справочная система туристического агентства
2. Информационно-справочная система ЖД вокзала
3. Информационно-справочная система аэропорта
4. Информационно-справочная система ресторана
5. Информационно-справочная система библиотеки

6. Информационно-справочная система магазина цифровой техники
7. Информационно-справочная система обработки и доставки заказов пиццерии
8. Информационно-справочная система кинотеатра
9. Информационно-справочная система театра

Примеры тестовых заданий

Тест 1. Результатом работы данного фрагмента кода будет

```
for(;;) {  
}
```

-: ошибка на этапе компиляции

-: ошибка на этапе выполнения

+: бесконечный цикл

-: ЭТОТ КОД НИКОГДА НЕ ВЫПОЛНИТСЯ

-: компилятор удалит лишнюю точку с запятой

Тест 2. Задан код

```
char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.' };  
String helloString = new String(helloArray);  
System.out.println ( helloString );
```

в результате работы программы в консоль будет выведено

-: h e l l o .

-: [h, e, l, l, o, .]

+: hello.

-: hello

-: ['h', 'e', 'l', 'l', 'o', '.']

Тест 3. В результате запуска кода

```
public class Main {  
    public static void main(String[] args) {  
        ArrayList<Integer> list;
```

```

    for (int i = 0; i < 10; i++){
        list.add(i);
    }
    System.out.println(list);
}
}

```

в консоль будет выведено

-: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

-: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

-: [0, 9]

+: java: variable list might not have been initialized

-: java: ';' expected

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

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

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

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

| Наименование компетенции | Наименование индикаторов достижения компетенции | Результаты обучения (умения и знания), соотношенные с индикаторами достижения компетенции | Типовые контрольные задания |
|--------------------------|---|---|-----------------------------|
|--------------------------|---|---|-----------------------------|

| | | | |
|--|---|---|---|
| ПКН-1. Способность применять общенаучные, общетехнические знания, математические методы в сфере ИТ | 1. Демонстрирует знания о современных естественнонаучных концепциях, общетехнических подходах, методах математического анализа и моделирования. | <p>Знать современные естественно-научные концепции, возможности аналитических информационных систем, общетехнические подходы и методы математического анализа и моделирования в рамках разработки программного обеспечения.</p> <p>Уметь применять информационные технологии для математического анализа и моделирования при разработке программного обеспечения.</p> | <p>Описать и объяснить основные принципы ООП. Объяснить математическую компоненту искусственного интеллекта в виде глубокого обучения и пояснить способы реализации нейронных сетей на языке программирования Java.</p> <p>Разработать информационную систему с реализацией различных статистических функций.</p> |
| | 2. Применяет знания для теоретического и экспериментального исследования в сфере разработки программного обеспечения. | <p>Знать современные методы теоретического и экспериментального исследования при управлении и разработкой проектов в области программного обеспечения.</p> <p>Уметь применять современные гибкие методы разработки программного обеспечения, включающие задачи планирования, анализа требований, проектирования, программирования, тестирования и документирования; осуществлять выбор программных и инструментальных средств для разработки, создания и отладки программного обеспечения; использовать готовые</p> | <p>Разработать проект по восстановлению данных во временных рядах посредством применения математического метода k-ближайших соседей.</p> <p>Разработать информационную систему по управлению научно-исследовательскими проектами с использованием ООП, JPA и технологии Hibernate.</p> |

| | | | |
|--|--|--|--|
| | | программные решения; организовывать верификацию, тестирование и проверку стабильности работы программного обеспечения. | |
|--|--|--|--|

Примеры практико-ориентированных (ситуационных) заданий

1. Создать JavaFX-приложение с формой авторизации – логином и паролем. Выполните документирование на основе Javadoc.

2. Создать на основе Spring MVC приложения с конфигурацией на основе аннотаций, которое выводит в браузер: Hello World. Написать тест к этому приложению на основе JUnit

3. Разработать калькулятор, реализующий операции сложения, вычитания, умножения, деления чисел любого типа. Интерфейс программы выполнить на JavaFX.

4. Разработать приложение с графическим интерфейсом JavaFX, в котором пользователь копирует в поле любой текст. Затем выводится список слов этого текста, рядом с каждым числом – сколько раз это слово встречается в тексте.

5. Написать программу вычисления объема параллелепипеда с графическим интерфейсом на основе JavaFX. Написать тест к этому приложению на основе JUnit 5.

6. Написать программу вычисления сопротивления электрической цепи, состоящей из двух последовательно соединенных сопротивлений. Написать тест к этому приложению на основе JUnit 5. Приложение должно быть с графическим интерфейсом JavaFX.

7. Написать на основе Spring Boot клиент-серверное приложение MyUser, в котором можно управлять данными пользователей из базы данных: имя, фамилия, возраст, номер группы. База данных может быть любой – MySQL, PostgreSQL и т.д.

8. Написать на основе Spring Boot Security форму для авторизации пользователя.

9. Напишите программу, которая вычисляет среднее арифметическое чисел, находящихся в файле, который указывает пользователь. Приложение должно быть с графическим интерфейсом JavaFX.

10. Напишите программу, которая выводит на экран список файлов, находящихся в указанном пользователем каталоге. Приложение должно быть с графическим интерфейсом JavaFX. Выполните документирование на основе Javadoc.

11. Напишите программу, которая создает и выводит на экран 5 случайных целых чисел. Диапазон чисел задает пользователь. Приложение должно быть с графическим интерфейсом JavaFX. Выполните документирование на основе Javadoc.

12. Напишите программу, которая выводит на экран все положительные двузначные числа (в возрастающем порядке). Диапазон чисел задает пользователь. Приложение должно быть с графическим интерфейсом JavaFX. Написать тест к этому приложению на основе JUnit 5.

13. Разработка MVC-приложения арифметический калькулятор на основе Spring Boot. Применить шаблонизатор Thymeleaf.

14. Написать программу, которая вычисляет сумму первых n членов ряда: $1+1/2+1/4+\dots$. Количество суммируемых членов ряда задается пользователем. Написать тест к этому приложению на основе JUnit 5. Приложение должно быть с графическим интерфейсом JavaFX.

15. Разработать программу, которая выводит на экран в строку все положительные двузначные четные числа (в возрастающем порядке). Диапазон вводит пользователь. Написать тест к этому приложению на основе JUnit 5. Приложение должно быть с графическим интерфейсом JavaFX.

16. Написать программу, которая вычисляет факториал введенного числа. Написать тест к этому приложению на основе JUnit 5. Приложение должно быть с графическим интерфейсом JavaFX. Выполните документирование на основе Javadoc.

17. Разработать программу, которая выводит положительные двузначные числа, делящиеся на 5 (в убывающем порядке). Диапазон вводит пользователь.

Написать тест к этому приложению на основе JUnit 5. Приложение должно быть с графическим интерфейсом JavaFX. Выполните документирование на основе Javadoc.

18. Создать JavaFX-приложение с формой авторизации – логином и паролем, а также пользовательской иконкой.

19. Разработка web-MVC приложения на основе Spring Boot. Приложение должно генерировать последовательность из 10 случайных чисел в диапазоне, заданном пользователем, и выводит эти числа на экран и вычисляет их среднее арифметическое.

20. Напишите программную реализацию бинарного дерева поиска. Приложение должно быть с графическим интерфейсом JavaFX.

21. Написать программу, который при введении даты в формате ДД.ММ.ГГ (к примеру, 22.10.20) выводит номер недели. Приложение должно быть с графическим интерфейсом JavaFX. Выполните документирование на основе Javadoc.

22. Разработать программу для выполнения операций сложения, вычитания и умножения комплексных чисел. Приложение должно быть с графическим интерфейсом JavaFX. Выполните документирование на основе Javadoc.

23. Реализуйте класс матрицы и методы

- Сложение и вычитание матриц.
- Умножение матрицы на число.
- Произведение двух матриц.
- Транспонированная матрица.
- Возведение матрицы в степень.
- Если метод, возвращает матрицу, то он должен возвращать новый объект, а не менять базовый.

Примерные вопросы для подготовки к зачету

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

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.

Пример экзаменационного билета

1 вопрос (20 баллов)

Виды контейнеров в Swing.

2 вопрос (20 баллов)

Spring Inversion of Control (IoC) контейнер Spring

3 вопрос (20 баллов)

Разработка MVC-приложения арифметический калькулятор на основе Spring Boot. Применить шаблонизатор Thymeleaf.

8. Перечень основной и дополнительной 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> (дата обращения: 19.12.2023). — Текст : электронный.

Дополнительная литература:

2. 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> (дата обращения: 19.12.2023). — Текст : электронный.

3. 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> (дата обращения: 19.12.2023). - Текст : электронный.

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

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система «Университетская библиотека ОН-ЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система Znanium <http://www.znanium.com>
5. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
6. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
7. Электронно-библиотечная система издательства Лань <https://e.lanbook.com/>
8. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
9. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>
10. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
11. Национальная электронная библиотека <http://нэб.рф/>
12. Финансовая справочная система «Финансовый директор» <http://www.1fd.ru/>
13. Java документация. <https://docs.oracle.com/en/java/>
14. The Java™ Tutorials <https://docs.oracle.com/javase/tutorial/java/concepts/index.html>
15. GoLang : GoLang - официальный сайт : сайт. – URL: <https://go.dev/> (дата обращения: 30.11.2023)
16. GIN - официальная документация : сайт. – URL: <https://gin-gonic.com/> (дата обращения: 30.11.2023)
17. FastAPI - официальная документация : сайт. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 30.11.2023)

10. Методические указания для обучающихся по освоению дисциплины

Освоение дисциплины складывается из усвоения теоретического материала и приобретения навыков разработки программ. Для изучения теоретического материала рекомендуется использовать тексты лекций и рекомендуемую литературу. Для успешного решения контрольных задач обучаемый должен проанализировать заданную предметную область, выделить объекты и разработать ее информационную модель. Разработка программы должна начинаться с составления логической схемы алгоритма (блок-схемы).

В начале семестра студенты должны самостоятельно с использованием зачетной книжки зарегистрироваться на сайте <https://www.jetbrains.com> и бесплатно получить IDE IntelliJ Idea, Free Educational Licenses или IntelliJ Idea Community Edition.

При переходе к новой теме проводится тестирование, направленное на оценивание теоретических знаний. Помимо тестирования, может проводиться выборочный устный опрос студентов. Полученные оценки участвуют в формировании итоговой оценки по дисциплине.

Практические навыки оцениваются путем разработки прикладных программ в визуальной среде программирования или разбора готовых программ. Студенты должны самостоятельно и вовремя решать поставленные преподавателем задачи.

Методические рекомендации по выполнению курсовой работы [8.8]: требуется разработать кроссплатформенное приложение (Клиент) и приложение, которое занимается хранением и обработкой информации, (Сервер), выполняющее учет и отображение объектов предметной области, хранящихся в базе данных (БД), и предоставляющее справочную и статистическую информацию. Информационная модель предметной области должна быть не проще двухуровневой структуры данных типа «родитель-дочка».

Клиент должен быть с графическим интерфейсом. Реализация допускается на языке программирования Java (графический интерфейс реализуется при помощи JavaFX). Клиент должен уметь соединяться с сервером при помощи запросов к API Сервера.

Сервер может не иметь интерфейса. Задача сервера, обрабатывать данные, полученные от Клиента, сохранять их в БД, делать запросы к БД. Сервер должен иметь реализацию RESTful API для взаимодействия с клиентом.

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. Пакет офисных программ.
2. Java Development Kit.
3. IDE IntelliJ Idea.
4. Антивирус Kaspersky.

11.2. Современные профессиональные базы данных и информационные справочные системы:

1.API документация JDK: <https://docs.oracle.com/en/java/javase/19/docs/api/index.html>

2.Руководство по JDK: <https://docs.oracle.com/en/java/javase/19/books.html>

11.3. Сертифицированные программные и аппаратные средства защиты информации: - не предусмотрены

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

Для проведения лекций и практических занятий необходима аудитория, оснащенная проектором и компьютерами с постоянным подключением к сети Интернет.