

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

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

**УТВЕРЖДАЮ**

**Проректор по учебной и  
методической работе**

\_\_\_\_\_ **Е.А. Каменева**

**24.12.2024 г.**

**Е.В. Пальчевский**

**Современные технологии программирования**

**Рабочая программа дисциплины**

**для студентов, обучающихся по направлению подготовки:**

**38.03.05 – Бизнес-информатика,**

**ОП «Цифровая трансформация управления бизнесом»**

*Рекомендовано Ученым советом  
Факультета информационных технологий и анализа больших данных  
(протокол № 50 от 17.12.2024 г.)*

*Одобрено советом Кафедры информационных технологий  
(протокол № 5 от 10.12.2024 г.)*

**Москва 2024**

## 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.....	6
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	6
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	6
5.1. Содержание дисциплины.....	6
5.2. Учебно-тематический план.....	9
5.3. Содержание семинаров, практических занятий.....	10
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	12
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	12
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю.....	13
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	15
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.....	23
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	24
10. Методические указания для обучающихся по освоению дисциплины.....	26
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем.....	26
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	27

## 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
УК-4	Способность использовать прикладное программное обеспечение при решении профессиональных задач	1. Использует основные методы и средства получения, представления, хранения и обработки данных.	<b>Знать:</b> принципы ООП, основные структуры данных (HashSet, HashMap), работу с форматом данных (JSON, XML), основы JDBC, методы сериализации и алгоритмы обработки данных. <b>Уметь:</b> получать, представлять, хранить и обрабатывать данные с использованием коллекций, Stream API, сериализации, а также работать с файлами и базами данных MySQL через JDBC.
		2. Демонстрирует владение профессиональными пакетами прикладных программ.	<b>Знать:</b> возможности и функционал профессиональных прикладных программ, их назначение и области применения, основы работы с офисными пакетами, инструментами для обработки данных, графики и программирования. <b>Уметь:</b> эффективно использовать инструменты для разработки ПО (например, IntelliJ IDEA, Eclipse), с целью решения профессиональных задач.
		3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	<b>Знать:</b> классификацию и назначение прикладного программного обеспечения, основные функции и возможности офисных, инженерных, графических и специализированных программ.

			<b>Уметь:</b> анализировать задачу, определять требования к программному обеспечению и выбирать подходящее приложение в зависимости от поставленной цели.
		4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.	<b>Знать:</b> функциональные возможности прикладного программного обеспечения, алгоритмы работы с ним и особенности применения в профессиональной деятельности. <b>Уметь:</b> применять специализированные программы для выполнения прикладных задач в зависимости от их специфики.
<b>УК-10</b>	Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации	<b>Знать:</b> методы описания структуры данных, виды и форматы данных, способы их сбора, хранения и обработки, а также основы интерпретации полученной информации. <b>Уметь:</b> формулировать требования к данным, разрабатывать их структуру, организовывать процессы сбора (с использованием баз данных, API, файловых систем), обработки (через анализ, сортировку, фильтрацию) и представлять результаты в удобной и понятной форме (графики, отчеты, визуализации).
		2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности	<b>Знать:</b> основные принципы, объясняющие сущность информационных процессов, методы выявления закономерностей для анализа данных. <b>Уметь:</b> анализировать данные, выявлять причинно-следственные связи, интерпретировать закономерности, оценивать природу и степень вариабельности с использованием статистических методов и инструментов обработки данных.
		3. Формулирует признак классификации, выделяет соответствующие ему группы	<b>Знать:</b> принципы и методы классификации, критерии группировки однородных

		<p>однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.</p>	<p>объектов, способы выявления общих свойств элементов групп и методы оценки полноты классификации.</p> <p><b>Уметь:</b> формулировать признаки классификации, выделять группы однородных объектов, идентифицировать общие свойства элементов, оценивать полноту результатов классификации и демонстрировать прикладные навыки назначения классификационных групп.</p>
		<p>4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.</p>	<p><b>Знать:</b> принципы логического мышления и аргументации, методы критического анализа информации, различия между фактами, мнениями, интерпретациями и оценками, а также основы построения обоснованных суждений и выводов.</p> <p><b>Уметь:</b> грамотно, логично и аргументированно формулировать собственные суждения и оценки; отличать факты от мнений, интерпретаций и оценок в рассуждениях других участников; применять критическое мышление для анализа информации и аргументов; обоснованно оценивать точки зрения и предоставлять конструктивную обратную связь.</p>
		<p>5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.</p>	<p><b>Знать:</b> принципы логического построения аргументов, методы системного описания и структурирования информации, правила эффективной коммуникации и убеждающей презентации, а также основы критического мышления и анализа.</p> <p><b>Уметь:</b> аргументированно и логично представлять свою точку зрения посредством системного описания, использовать структурированные подходы для обоснования своих позиций, ясно и последовательно излагать мысли, подкрепляя их фактами и логическими связками.</p>

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Современные технологии программирования» относится к Циклу математики и информатики по направлению подготовки 38.03.05 – Бизнес-информатика, ОП «Цифровая трансформация управления бизнесом».

### 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 2 (в часах)
<b>Общая трудоемкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<b>Контактная работа – Аудиторные занятия</b>	<b>34</b>	<b>34</b>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>18</i>	<i>18</i>
<b>Самостоятельная работа</b>	<b>74</b>	<b>74</b>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

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

#### 5.1. Содержание дисциплины

#### Тема 1. 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**

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.

## **Тема 3. Базовые принципы объектно-ориентированного программирования**

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.

## **Тема 4. Классы, методы и объекты Java**

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.

## **Тема 5. Интерфейсы. Исключения. Отладка программ.**

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 6. Коллекции.**

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 7. Потоквая 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. Потоквая 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 8. Работа с базами данных в Java. Основы объектно-реляционного отображения.**

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. За-



просы 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.

## 5.2. Учебно-тематический план

№ 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	13	4	2	2	9	Тестовые задания (контрольная ра- бота), устный опрос, проверка практических зада- ний, онлайн-курсы (п.9)
2	Основные язы- ковые конструк- ции Java	13	4	2	2	9	Тестовые задания (контрольная ра- бота), устный опрос, проверка практических зада- ний, онлайн-курсы (п.9)
3	Базовые прин- ципы объектно- ориентирован- ного программи- рования	13	4	2	2	9	Тестовые задания (контрольная ра- бота), устный опрос, проверка практических зада- ний, онлайн-курсы (п.9)
4	Классы, методы и объекты Java	13	4	2	2	9	Тестовые задания (контрольная ра- бота), устный опрос, проверка практических зада- ний, онлайн-курсы (п.9)

5	Интерфейсы. Исключения. Отладка программ	13	4	2	2	9	Тестовые задания (контрольная работа), устный опрос, проверка практических заданий, онлайн-курсы (п.9)
6	Коллекции	13	4	2	2	9	Тестовые задания (контрольная работа), устный опрос, проверка практических заданий, онлайн-курсы (п.9)
7	Потоковая организация системы ввода-вывода. Многопоточность	13	4	2	2	9	Тестовые задания (контрольная работа), устный опрос, проверка практических заданий, онлайн-курсы (п.9)
8	Работа с базами данных в Java. Основы объектно-реляционного отображения.	17	6	2	4	11	Тестовые задания (контрольная работа), устный опрос, проверка практических заданий, онлайн-курсы (п.9)
	В целом по дисциплине	108	34	16	18	74	Согласно учебному плану: контрольная работа
	Итого в %		31	47	53	69	

\* 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	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	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа

	<p>строки. Прimitives и ссылочные типы данных, литералы. Операции. Преобразование типов. Переменные: объявление, инициализация.</p> <p><i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i></p>	
Основные языковые конструкции Java	<p>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.</p> <p><i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i></p>	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа
Базовые принципы объектно-ориентированного программирования	<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><i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i></p>	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа
Классы, методы и объекты Java	<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><i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i></p>	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа
Интерфейсы. Исключения. Отладка программ	<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, 9.23 – 9.27</i></p>	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа
Коллекции	<p>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.</p> <p><i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i></p>	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа
Потоковая организация системы ввода-вывода. Многопоточность.	<p>Консольный ввод, запись консольного вывода. Классы-реализации потоков данных. Классы ByteArrayInputStream и ByteArrayOutputStream. Классы FileInputStream и FileOutputStream. StringBufferInputStream. Классы FilterInputStreem и FilterOutputStream. BufferedInputStream и BufferedOutputStream. DataInputStream и DataOutputStream. Класс Thread и интерфейс Runnable. Жизненный цикл потока. Фоновые задачи. Управление потоками и приоритеты потоков.</p>	Дискуссия по результатам самостоятельной работы, решение задач, аудиторная проверочная работа

	Синхронизация. Приостановка, возобновление и остановка потока. <i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i>	
Работа с базами данных в Java. Основы объектно-реляционного отображения.	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. <i>Рекомендуемые источники: 8.1, 8.2, 9.23 – 9.27</i>	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, прохождение онлайн-курсов
Основные языковые конструкции Java	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, прохождение онлайн-курсов
Базовые принципы объектно-ориентированного программирования	Поиск пакета и переменная среды CLASSPATH.	Обзор литературы и веб-источников, прохождение онлайн-курсов
Классы, методы и объекты Java	Защищенные члены классов. Класс Object. Многоуровневая иерархия классов. Передача объектов методам. Инициализация полей по умолчанию. Вызов одного конструктора из другого.	Обзор литературы и веб-источников, прохождение онлайн-курсов
Интерфейсы. Исключения. Отладка программ	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, прохождение онлайн-курсов
Коллекции	Связные списки. Хэш-множества. Очереди по приоритету. Древовидные множества. Взаимные	Обзор литературы и веб-источников, прохождение онлайн-курсов

	преобразования списков и множеств. Стеки. Битовые множества.	
Потоковая организация системы ввода-вывода. Многопоточность.	Классы PipedInputStream и PipedOutputStream. SequenceInputStream. LineNumberInputStream. PushBackInputStream. PrintStream. Синхронизированные блоки. Параллельное программирование.	Обзор литературы и веб-источников, прохождение онлайн-курсов
Работа с базами данных в Java. Основы объектно-реляционного отображения.	Исключения (SQL Exception). Потоки в JDBC. Основные этапы работы с базой данных при использовании JDBC. Уровни изоляции транзакций в JDBC. Хранимая процедура.	Обзор литературы и веб-источников, прохождение онлайн-курсов

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

### Примерный вариант контрольной работы

#### 1. Тема: «Базовые знания Java»

##### Задание 1

Напишите методы для обработки коллекций и строк.

##### Задание 2

Реализуйте классы и интерфейсы, включая абстрактные классы.

##### Задание 3

Используйте рефлексию для инспекции и модификации объектов на уровне выполнения.

#### 2. Тема: «Работа с базой данных»

##### Задание 1

Подключитесь к базе данных MySQL через JDBC.

##### Задание 2

Создайте простую таблицу пользователей с использованием SQL и выполните операции вставки, обновления и удаления данных.

##### Задание 3

Импортируйте данные из файла CSV в базу данных.

## Примеры тестовых заданий

**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

**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', '.']

**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 Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.*

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

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

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

<b>Наименование компетенции</b>	<b>Наименование индикаторов достижения компетенции</b>	<b>Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции</b>	<b>Типовые контрольные задания</b>
УК-4. Способность использовать прикладное программное обеспечение при решении профессиональных задач	1. Использует основные методы и средства получения, представления, хранения и обработки данных.	<b>Знать:</b> принципы ООП, основные структуры данных (HashSet, HashMap), работу с форматом данных (JSON, XML), основы JDBC, методы сериализации и алгоритмы обработки данных. <b>Уметь:</b> получать, представлять, хранить и обрабатывать данные с	Разработайте на языке Java консольное приложение, которое считывает данные из JSON-файла (например, список товаров), представляет их с использованием классов и коллекций (HashSet, HashMap), обрабатывает данные с применением Stream API (фильтрация, сортировка), сохраняет объекты посредством сериализации в файл,

		использованием коллекций, Stream API, сериализации, а также работать с файлами и базами данных MySQL через JDBC.	а также записывает данные в базу данных MySQL через JDBC; приложение должно демонстрировать знания принципов ООП, работы с форматами JSON и XML, методов сериализации и алгоритмов обработки данных.
	2. Демонстрирует владение профессиональными пакетами прикладных программ.	<b>Знать:</b> возможности и функционал профессиональных прикладных программ, их назначение и области применения, основы работы с офисными пакетами, инструментами для обработки данных, графики и программирования. <b>Уметь:</b> эффективно использовать инструменты для разработки ПО (например, IntelliJ IDEA, Eclipse), с целью решения профессиональных задач.	Разработайте небольшой проект на Java с использованием среды разработки IntelliJ IDEA или Eclipse, в котором реализованы основные операции обработки данных (чтение, обработка, вывод) с применением встроенных инструментов IDE (отладка, тестирование, работа с библиотеками). В ходе выполнения задания необходимо продемонстрировать навыки эффективной настройки среды, использования встроенных средств анализа кода, работы с инструментами интеграции систем контроля версий (например, Git), а также показать умение решать профессиональные задачи с использованием возможностей IDE.
	3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	<b>Знать:</b> классификацию и назначение прикладного программного обеспечения, основные функции и возможности офисных, инженерных, графических и специализированных программ. <b>Уметь:</b> анализировать задачу, определять требования к программному обеспечению и выбирать подходящее приложение в зависимости от поставленной цели.	Вам необходимо разработать на языке Java приложение для управления заказами в интернет-магазине. Проанализируйте поставленную задачу, определите требования к программному обеспечению и выберите подходящие инструменты и библиотеки для разработки Java-приложения, обосновав свой выбор. Например, для разработки используйте интегрированную среду разработки <b>IntelliJ IDEA</b> , для работы с базами данных — <b>MySQL</b> вместе с JDBC. Опишите основные функции и возможности выбранных программ и библиотек, которые помогут эффективно реализовать поставленную задачу.
	4. Использует прикладное программное обеспечение	<b>Знать:</b> функциональные возможности прикладного программного обеспечения,	Вам необходимо разработать на языке Java приложение для ана-



	для решения конкретных прикладных задач.	<p>алгоритмы работы с ним и особенности применения в профессиональной деятельности.</p> <p><b>Уметь:</b> применять специализированные программы для выполнения прикладных задач в зависимости от их специфики.</p>	<p>лиза данных о продажах компании, используя интегрированную среду разработки IntelliJ IDEA для кодирования и библиотеку Apache POI для работы с Excel-файлами; опишите функциональные возможности этих инструментов, алгоритмы их работы и особенности применения в профессиональной деятельности, затем реализуйте программу, которая считывает данные из Excel, обрабатывает их (например, вычисляет общие продажи по категориям или периодам) и выводит результаты в консоль или сохраняет их в новый Excel-файл, демонстрируя умение применять специализированные программы для решения прикладных задач в зависимости от их специфики.</p>
УК-10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации	<p><b>Знать:</b> методы описания структуры данных, виды и форматы данных, способы их сбора, хранения и обработки, а также основы интерпретации полученной информации.</p> <p><b>Уметь:</b> формулировать требования к данным, разрабатывать их структуру, организовывать процессы сбора (с использованием баз данных, API, файловых систем), обработки (через анализ, сортировку, фильтрацию) и представлять результаты в удобной и понятной форме (графики, отчеты, визуализации).</p>	<p>Разработайте на языке Java консольное приложение для управления списком студентов. Приложение должно выполнять следующие функции: считывать данные о студентах из текстового файла (например, имя, фамилия, возраст, группа), хранить их в подходящей структуре данных (например, ArrayList или HashMap), выполнять обработку данных путем сортировки списка по фамилии и фильтрации студентов по определенной группе, а также выводить результаты в удобном текстовом формате на экран или записывать в новый файл. В задании необходимо описать методы описания структуры данных, виды используемых форматов данных, способы сбора и хранения информации, а также продемонстрировать умение анализировать, сортировать и фильтровать данные, представляя результаты в понятной форме без использования API и графической визуализации.</p>

	<p>2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности</p>	<p><b>Знать:</b> основные принципы, объясняющие сущность информационных процессов, методы выявления закономерностей для анализа данных.</p> <p><b>Уметь:</b> анализировать данные, выявлять причинно-следственные связи, интерпретировать закономерности, оценивать природу и степень вариабельности с использованием статистических методов и инструментов обработки данных.</p>	<p>Разработайте на языке Java консольное приложение для анализа данных о продажах магазина. Приложение должно считывать данные из текстового файла, содержащего информацию о каждой продаже (например, дата, товар, количество, цена). Реализуйте функционал для анализа данных, включающий вычисление общей суммы продаж, выявление наиболее продаваемых товаров, определение закономерностей в продажах по дням недели, а также оценку вариабельности продаж с использованием статистических методов (например, расчет среднего значения, медианы и стандартного отклонения). Программа должна выводить результаты анализа в консоль, демонстрируя умение анализировать данные, выявлять причинно-следственные связи и закономерности, а также оценивать природу и степень вариабельности с использованием статистических методов и инструментов обработки данных.</p>
	<p>3. Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.</p>	<p><b>Знать:</b> принципы и методы классификации, критерии группировки однородных объектов, способы выявления общих свойств элементов групп и методы оценки полноты классификации.</p> <p><b>Уметь:</b> формулировать признаки классификации, выделять группы однородных объектов, идентифицировать общие свойства элементов, оценивать полноту результатов классификации и демонстрировать прикладные навыки назначения классификационных групп.</p>	<p>Разработайте на языке Java консольное приложение для классификации студентов по специальностям. Приложение должно считывать данные из текстового файла, содержащего информацию о студентах (например, имя, возраст, оценки, выбранная специальность). Реализуйте функционал для формулирования признаков классификации (например, оценки и предпочтения), выделения групп однородных студентов по специальностям, идентификации общих свойств внутри каждой группы (например, средний балл, возрастной диапазон) и оценки полноты классификации (насколько все студенты правильно распределены по специальностям).</p>

			Программа должна выводить результаты классификации в консоль, демонстрируя умение применять принципы и методы классификации, критерии группировки, выявлять общие свойства элементов групп и оценивать полноту выполненной классификации.
	4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	<p><b>Знать:</b> принципы логического мышления и аргументации, методы критического анализа информации, различия между фактами, мнениями, интерпретациями и оценками, а также основы построения обоснованных суждений и выводов.</p> <p><b>Уметь:</b> грамотно, логично и аргументированно формулировать собственные суждения и оценки; отличать факты от мнений, интерпретаций и оценок в рассуждениях других участников; применять критическое мышление для анализа информации и аргументов; обоснованно оценивать точки зрения и предоставлять конструктивную обратную связь.</p>	Разработайте на языке Java консольное приложение для управления списком сотрудников компании. Приложение должно выполнять следующие функции: считывать данные о сотрудниках из текстового файла (например, имя, должность, возраст, зарплата), хранить их в подходящей структуре данных (например, ArrayList или HashMap), выполнять обработку данных путем сортировки по зарплате и фильтрации по должности, а также выводить результаты на экран или сохранять в новый файл. В процессе разработки подготовьте краткий отчет, в котором аргументированно опишите принятые решения по выбору структур данных и алгоритмов, различая фактические преимущества и личные предпочтения. Таким образом, вы продемонстрируете знание принципов логического мышления и аргументации, методы критического анализа информации, а также умение формулировать обоснованные суждения и отличать факты от мнений в процессе разработки программного обеспечения.
	5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	<p><b>Знать:</b> принципы логического построения аргументов, методы системного описания и структурирования информации, правила эффективной коммуникации и убеждающей презентации, а также основы критического мышления и анализа.</p>	Разработайте на языке Java консольное приложение для управления библиотечным каталогом (например, добавление, удаление, поиск и отображение книг). После реализации приложения подготовьте письменный отчет, в котором аргументированно и логично опишите выбранную

		<p><b>Уметь:</b> аргументированно и логично представлять свою точку зрения посредством системного описания, использовать структурированные подходы для обоснования своих позиций, ясно и последовательно излагать мысли, подкрепляя их фактами и логическими связками.</p>	<p>структуру программы, обоснуйте решения по организации кода и использованию определённых методов и классов. В отчёте используйте системное описание и структурированные подходы для обоснования своих позиций, чётко и последовательно излагайте мысли, подкрепляя их фактами и логическими связками. Таким образом, вы продемонстрируете знание принципов логического построения аргументов, методов системного описания и структурирования информации, правил эффективной коммуникации и основы критического мышления и анализа.</p>
--	--	--	--

### ***Примеры практико-ориентированных (ситуационных) заданий***

**1. Базовый вариант.** Строковый тип данных (класс StringBuffer). Ввести две строки (не менее 50 символов каждая) с клавиатуры. Необходимо вывести на экран две введенных ранее строки, изменить порядок символов на обратный и добавить одну строку в другую. **На основе базового реализовать сложный вариант.** Реализовать программу с интерактивным консольным меню, (т.е. вывод списка действий по цифрам. При этом при нажатии на цифру у нас должно выполняться определенное действие). Задачи полностью идентичны базовому варианту. 1. Вывести все таблицы из MySQL. 2. Создать таблицу в MySQL. 3. Изменить порядок символов строки на обратный, результат сохранить в MySQL с последующим выводом в консоль. 4. Добавить одну строку в другую, результат сохранить в MySQL с последующим выводом в консоль. 5. Сохранить все данные (выше полученные результаты) из MySQL в Excel и вывести на экран.

**2. Базовый вариант.** Объектно-ориентированное программирование. Необходимо создать класс ArrayPI (модификатор доступа – public), в котором необходимо создать два двумерных массива (ввод с клавиатуры, 7 столбцов и 7 строк). Далее необходимо создать класс-наследник Matrix (модификатор доступа – public

final), в котором будут наследоваться данные матрицы и перемножаться. Необходимо перемножить две данных матрицы и итоговую матрицу (произведение двух матриц) вывести на экран. **На основе базового реализовать сложный вариант.** Реализовать программу с интерактивным консольным меню, (т.е. вывод списка действий по цифрам. При этом при нажатии на цифру у нас должно выполняться определенное действие). Задачи полностью идентичны базовому варианту. Каждый пункт меню должен быть отдельным классом-наследником(подклассом). 1. Вывести все таблицы из базы данных MySQL. 2. Создать таблицу в базе данных MySQL. 3. Ввести две матрицы с клавиатуры и каждую из них сохранить в MySQL с последующим выводом в консоль. 4. Перемножить матрицу, сохранить перемноженную матрицу в MySQL и вывести в консоль. 5. Сохранить результаты из MySQL в Excel и вывести их в консоль.

#### ***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.

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

### **Основная литература:**

- 8.1. Гуськова, О. И. Объектно ориентированное программирование в Java: учебное пособие / О. И. Гуськова. - Москва: МПГУ, 2018. - 240 с. - ЭБС ZNANIUM. - URL: <https://znanium.com/catalog/product/1020593> (дата обращения: 13.11.2024). – Текст: электронный.
- 8.2. 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> (дата обращения: 13.11.2024). — Текст: электронный.

### **Дополнительная литература:**

8.3. 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> (дата обращения: 13.11.2024). — Текст: электронный.

### **9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплин**

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
2. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
3. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
4. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
5. Электронно-библиотечная система Znanium <http://www.znaniy.com>
6. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
7. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
8. Электронно-библиотечная система издательства Лань <https://e.lanbook.com/>
9. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
10. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>
11. Математические журналы: полнотекстовая коллекция Математического института им. В.А. Стеклова РАН <https://www.mathnet.ru/>
12. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
13. Национальная электронная библиотека <http://нэб.рф/>
14. Ресурсы информационно-аналитического агентства по финансовым рынкам Cbonds.ru <https://cbonds.ru/>
15. СПАРК <https://spark-interfax.ru/>
16. CNKI. Academic Reference <https://ar.oversea.cnki.net/>
17. Электронные продукты издательства Elsevier <http://www.sciencedirect.com>



18. Emerald: Management eJournal Portfolio <https://www.emerald.com/insight/>
19. Реферативная база данных по математике MathSciNET <https://mathscinet.ams.org/mathscinet/>
20. Коллекция научных журналов Oxford University Press <https://academic.oup.com/journals/>
21. Электронные коллекции книг и журналов издательства Springer: <http://link.springer.com/>
22. Платформа STATISTA <https://www.statista.com/>
23. База данных научных журналов издательства Wiley <https://onlinelibrary.wiley.com/>
24. Java документация. <https://docs.oracle.com/en/java/>
25. The Java™ Tutorials <https://docs.oracle.com/javase/tutorial/java/concepts/index.html>
26. Бесплатные курсы в рамках Ассоциированной программы Яндекса на платформе Я.Практикум (освоение происходит в любой период времени в рамках прохождения дисциплины): <https://practicum.yandex.ru/professions/associated-programs-backend-st/subscribe> (Backend); <https://practicum.yandex.ru/professions/associated-programs-frontend-st/subscribe> (Frontend).
27. Бесплатные курсы на платформе «Stepik» (освоение происходит в любой период времени в рамках прохождения дисциплины):
  - «Основы Java»: <https://stepik.org/course/82867/promo;>
  - «Разработка веб сервиса на Java (часть 1)»: <https://stepik.org/course/146/promo;>
  - «Интерактивный тренажер по SQL»: <https://stepik.org/course/63054/promo;>
  - «Введение в базы данных»: [https://stepik.org/course/1240/promo.](https://stepik.org/course/1240/promo)
28. Курс на платформе «Campus Moodle» (освоение происходит в любой период времени в рамках прохождения дисциплины): <https://campus.fa.ru/course/view.php?id=10185>

## **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 Ultimate, 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.

## **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.