Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Любовь Алексеевна
Должность: Ректор
Дата подписания: 29.10.2026 15:29:52
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2c42baff9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Поволжский государственный университет сервиса» (ФГБОУ ВО «ПВГУС»)

Высшая школа интеллектуальных систем и кибертехнологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«ПРОГРАММИРОВАНИЕ»

Направление подготовки:
09.03.02 «Информационные системы и технологии»

Направленность (профиль):
«Информационные системы и технологии»

Квалификация выпускника: **бакалавр**

Тольятти 2025

Рабочая программа дисциплины «Программирование» разработана в соответствии с федеральным государственным образовательным стандартом высшего образования - бакалавриат по направлению подготовки 09.03.02 «Информационные системы и технологии», утвержденным приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. N 926.

Составители:

старший преподаватель Коновалова А.Д.
(ученая степень, ученое звание) (ФИО)

РПД обсуждена на заседании Высшей школы интеллектуальных систем и кибертехнологий

«23» 0 5 2025 г., протокол № 10

Директор ВШИСиКТ к.э.н., доцент Филиппова О.А.
(уч.степень, уч.звание) (ФИО)

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Цель освоения дисциплины

Целью освоения дисциплины является:

– формирование у обучающихся общепрофессиональных компетенций, обеспечивающих разработку алгоритмов и программ, пригодных для практического применения в области информационных систем и технологий.

– формирование профессиональных компетенций по проектированию программного обеспечения, анализу требований и разработке технических спецификаций.

1.2. Перечень планируемых результатов обучения по дисциплине

Код и наименование компетенции	**Код и наименование индикатора достижения компетенции**	**Планируемые результаты обучения по дисциплине**	**Основание (ПС)** *для профессиональных компетенций
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ИОПК-6.2. Использует современные языки программирования для разработки алгоритмов и программ ИОПК-6.3. Владеет методами отладки и тестирования программ	**Знает:** синтаксис и семантику основных конструкций современного языка программирования; способы представления алгоритмов; принципы модульной и объектно-ориентированной организации программ; методы отладки и тестирования. **Умеет:** разрабатывать алгоритмы, реализовывать их на языке программирования, организовывать программу из функций и модулей, применять структуры данных, находить и исправлять ошибки, проектировать тестовые случаи. **Владеет:** навыками написания, запуска, отладки, тестирования и документирования программ, пригодных для практического применения.	–
ПК-1 Способен к выполнению работ по проектированию программного обеспечения	ИПК-1.1. Осуществляет разработку требований к программному обеспечению и анализ исполнения требований ИПК-1.2. Выполняет разработку технических специфик	**Знает:** назначение и структуру требований к программному обеспечению, способы их анализа и трассировки; назначение технической спецификации и критериев приемки. **Умеет:** выявлять и формализовать функциональные и нефункциональные требования, анализировать их выполнимость и непротиворечивость, разрабатывать технические спецификации программных компонентов. **Владеет:** навыками декомпозиции требований, подготовки технических спецификаций и согласования требований с проектным решением.	06.001 «Программист».

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к части, формируемой участниками образовательных отношений, Блока 1. «Дисциплины (модули)» образовательной программы (Профессиональный модуль).

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объем и структура дисциплины

Общая трудоемкость дисциплины составляет **7 з.е. (252 час.)**, их распределение по видам работ и формам обучения представлено в таблице.

Виды учебных занятий и работы обучающихся	**Трудоемкость, час**		
	Всего	**3 семестр**	**4семестр**
Общая трудоемкость дисциплины, час	**252**	**108**	**144**
Контактная работа обучающихся с преподавателем по видам учебных занятий (всего), в т.ч.:	**96/28**	**44/10**	**52/18**
занятия лекционного типа (лекции)	34/10	16/4	18/6
занятия семинарского типа (семинары, практические занятия, практикумы, коллоквиумы и иные аналогичные занятия)	34/10	16/4	18/6
Лабораторные работы	28/8	12/2	16/6
Самостоятельная работа всего, в т.ч.:	**129/211**	**64/94**	**65/117**
Самоподготовка по темам (разделам) дисциплины	129/211	64/94	65/117
Выполнение курсового проекта /курсовой работы	36/36	-/-	36/36
Контроль (часы на экзамен, зачет)	**27/13**	**0/4**	**27/9**
Промежуточная аттестация		**Зачёт**	**Экзамен/ защита КР**

Примечание: *-/- объем часов соответственно для очной, очно-заочной форм обучения*

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

В процессе освоения дисциплины может применяться электронное обучение и дистанционные образовательные технологии.

В процессе освоения дисциплины обучающиеся обеспечены доступом к электронной информационно-образовательной среде и электронно-библиотечным системам.

3.1. Содержание дисциплины, структурированное по темам

Планируемые результаты освоения: код формируемой компетенции и индикаторы достижения компетенций	Наименование разделов, тем	Виды учебной работы				Формы текущего контроля (наименование оценочного средства)
		Контактная работа			Самостоятельная работа, час	
		Лекции, час	Лабораторные работы, час	Практические занятия, час		
3 семестр						
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 1. ВВЕДЕНИЕ В JAVA. СРЕДА РАЗРАБОТКИ. БАЗОВЫЙ СИНТАКСИС, ТИПЫ ДАННЫХ И ОПЕРАЦИИ** Содержание лекции: 1. Архитектура Java: JVM, JRE, JDK. Компиляция и интерпретация байт-кода. 2. Обзор сред разработки. Структура Java-программы: классы, метод main(). 3. Базовые примитивные типы. Ссылочные типы. Переменные, константы. 4. Операции: арифметические, сравнения, логические, побитовые, тернарный оператор. Преобразование и приведение типов. 5. Ввод/вывод данных: класс Scanner, System.out.println(). Форматированный вывод.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 1. Установка и настройка. Создание первого Java-проекта. Реализация линейных алгоритмов и вычисление арифметических выражений с использованием примитивных типов.		2/1			Отчет по лабораторной работе
	Практическая работа 1. Построение блок-схем алгоритмов. Разработка программ для решения задач с использованием арифметических операций и класса Math.			2/1		Отчёт по практической работе
	Самостоятельная работа.				10/15	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 2. УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ. УСЛОВНЫЕ ОПЕРАТОРЫ И ЛОГИКА В JAVA** Содержание лекции: 1. Логический тип boolean. 2. Операции сравнения и логические операторы. 3. Условный оператор if, if-else, множественный выбор с использованием switch. 4. Вложенные ветвления. Проверка предусловий. Обработка некорректных входных данных. 5. Использование метода equals() для сравнения строк.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 2. Реализация программ с условными операторами и оператором switch.		2/0			Отчет по лабораторной работе
	Практическая работа 2. Проектирование и проверка ветвящихся алгоритмов.			2/1		Отчёт по практической работе
	Самостоятельная работа.				10/15	Самостоятельное изучение учебных

						материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 3. ЦИКЛЫ В JAVA. ПОСЛЕДОВАТЕЛЬНОСТИ И НАКОПИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ** Содержание лекции: 1. Циклы в Java. Управление циклом. Счетчики и накопители. 2. Вложенные циклы. Генерация числовых последовательностей. 3. Циклы для обработки строк. Бесконечные циклы и контроль завершения. Итераторы.	2/0				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 3. Программирование циклических вычислений. Табулирование функций с использованием циклов.		2/0			Отчет по лабораторной работе
	Практическая работа 3. Трассировка циклов. Решение задач на накопление суммы и произведения.			2/0		Отчёт по практической работе
	Самостоятельная работа.				11/16	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 4. МАССИВЫ, МАТРИЦЫ И СТРОКИ В JAVA** Содержание лекции: 1. Массивы в Java: одномерные и многомерные. 2. Объявление, инициализация, заполнение. Индексация и обход массивов. 3. Алгоритмы поиска. Алгоритмы сортировки. Строки. 4. Классы StringBuilder и StringBuffer для работы с изменяемыми строками. Сравнение строк.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 4. Обработка массивов и матриц. Реализация алгоритмов сортировки. Обработка строк.		2/1			Отчет по лабораторной работе
	Практическая работа 4. Проектирование алгоритмов обработки массивов и строк.			4/1		Отчёт по практической работе
	Самостоятельная работа.				11/16	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 5. МЕТОДЫ (ФУНКЦИИ) И МОДУЛЬНАЯ ОРГАНИЗАЦИЯ ПРОГРАММЫ В JAVA** Содержание лекции: 1. Декомпозиция задачи на методы. Объявление методов. Передача параметров по значению. Возврат значений. Перегрузка методов. 2. Область видимости переменных. Статические и нестатические методы. Понятие рекурсивных методов. 3. Организация пакетов и импорт классов. Модификаторы доступа. 4. Принципы написания чистого кода в Java: именование, длина методов, комментарии.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 5. Разработка набора методов для решения предметной задачи. Объединение методов в пакеты. Вызов методов из главного класса.		2/0			Отчет по лабораторной работе

	Практическая работа 5. Декомпозиция задачи на методы. Проектирование интерфейсов методов.			4/1		Отчёт по практической работе
	Самостоятельная работа.				11/16	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 6. РАБОТА С ФАЙЛАМИ, СТРУКТУРАМИ ДАННЫХ И ИСКЛЮЧЕНИЯМИ В JAVA. ОСНОВЫ ОТЛАДКИ И ТЕСТИРОВАНИЯ** Содержание лекции: 1. Пользовательские типы данных. 2. Работа с текстовыми файлами. Работа с двоичными файлами. 3. Иерархия исключений в Java. Проверяемые и непроверяемые исключения. Конструкция try-catch-finally. Ресурсный блок try-with-resources. Создание собственных исключений. 4. Отладка в IntelliJ IDEA: точки останова, пошаговое выполнение, просмотр переменных.	2/0				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 6. Обработка структурированных данных из текстового файла. Реализация записи и чтения данных.		2/0			Отчет по лабораторной работе
	Практическая работа 6. Разработка тестовых наборов для программного модуля. Анализ дефектов и их устранение с использованием отладчика.			2/0		Отчёт по практической работе
	Самостоятельная работа.				11/16	Самостоятельное изучение учебных материалов
	Итого 3 семестр	**16/4**	**12/2**	**16/4**	**64/94**	
4 семестр						
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 7. РЕКУРСИЯ, ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ И ОЦЕНКА АЛГОРИТМОВ В JAVA** Содержание лекции: 1. Рекурсивные методы в Java. Стек вызовов и ограничения рекурсии. 2. Сравнение итеративных и рекурсивных решений. Динамические структуры данных: связные списки. 3. Основные операции со списками. Понятие вычислительной сложности. Сравнение алгоритмов по времени и памяти.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 7. Реализация и сравнение рекурсивных и итеративных алгоритмов для одной и той же задачи. Разработка базовых операций с односвязным списком.		2/1			Отчет по лабораторной работе
	Практическая работа 7. Построение рекурсивных алгоритмов и оценка их ограничений.			3/1		Отчёт по практической работе
	Самостоятельная работа.				10/19	Самостоятельное изучение учебных материалов
ОПК-6:	**ТЕМА 8. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ**	3/1				Лекция-

ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ПРОГРАММИРОВАНИЕ В JAVA: КЛАССЫ И ОБЪЕКТЫ** Содержание лекции: 1. Основы объектно-ориентированного программирования в Java. Классы и объекты. 2. Поля. Методы. Конструкторы. 3. Инкапсуляция: модификаторы доступа. Геттеры и сеттеры. Жизненный цикл объекта. 4. Ключевое слово this. Композиция и агрегация. Ассоциации между классами. Принципы проектирования простых классов.					визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 8. Разработка классов предметной области. Реализация методов обработки и вывода информации.		2/1			Отчет по лабораторной работе
	Практическая работа 8. Выделение объектов предметной области. Проектирование ответственности классов.			3/1		Отчёт по практической работе
	Самостоятельная работа.				11/19	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 9. НАСЛЕДОВАНИЕ, ПОЛИМОРФИЗМ И ИНТЕРФЕЙСЫ В JAVA** Содержание лекции: 1. Наследование. Повторное использование кода. Переопределение методов. 2. Полиморфизм. Абстрактные классы. 3. Интерфейсы: дефолтные и статические методы. Множественное наследование через интерфейсы. Принцип подстановки Лисков. Композиция vs наследование. 4. Инкапсуляция, наследование и полиморфизм – три кита ООП.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 9. Реализация иерархии классов. Полиморфная обработка коллекции объектов. Реализация интерфейсов Comparable и Comparator для сортировки.		2/1			Отчет по лабораторной работе
	Практическая работа 9. Анализ проектных решений с использованием наследования и интерфейсов. Проектирование системы с использованием абстрактных классов и интерфейсов.			3/1		Отчёт по практической работе
	Самостоятельная работа.				11/19	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 10. КОЛЛЕКЦИИ, ОБОБЩЕНИЯ И ФУНКЦИОНАЛЬНЫЕ СРЕДСТВА JAVA** Содержание лекции: 1. Java Collections Framework (JCF): иерархия. 2. Интерфейсы Collection, List, Set, Queue, Map. 3. Классы-реализации: ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap. 4. Итераторы. Цикл for-each. Обобщенные типы: параметризованные классы и методы. 5. Ограничения. Сравнение структур данных по времени выполнения операций. Функциональные интерфейсы. Лямбда-выражения.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 10. Обработка коллекций объектов с использованием типовых алгоритмов:		4/1			Отчет по лабораторной

	поиск, сортировка, фильтрация, группировка.					работе
	Практическая работа 10. Выбор оптимальной структуры данных для заданных операций.			3/1		Отчёт по практической работе
	Самостоятельная работа.				11/20	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 11. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ** Содержание лекции: 1. Источники требований. Функциональные и нефункциональные требования. Критерии качества требований. 2. Анализ реализуемости требований в контексте Java-стек технологий. Декомпозиция требований на модули. 3. Техническая спецификация программного компонента: назначение, интерфейсы, структуры данных, ограничения, критерии приемки.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 11. Преобразование набора требований в техническую спецификацию для Java-компонента.		4/1			Отчет по лабораторной работе
	Практическая работа 11. Анализ неоднозначных и противоречивых требований. Формирование критериев приемки и их связь с тест-кейсами.			3/1		Отчёт по практической работе
	Самостоятельная работа.				11/20	Самостоятельное изучение учебных материалов
ОПК-6: ИОПК-6.2 ИОПК-6.3 ПК-1: ИПК-1.1 ИПК-1.2	**ТЕМА 12. КОМПЛЕКСНАЯ РАЗРАБОТКА JAVA-ПРИЛОЖЕНИЯ: АРХИТЕКТУРА, ТЕСТИРОВАНИЕ, ДОКУМЕНТИРОВАНИЕ** Содержание лекции: 1. Сборка программного решения из модулей. Организация зависимостей с помощью Maven/Gradle. 2. Стратегии тестирования: модульное, интеграционное, регрессионное. 3. Документирование кода. Документирование пользовательских сценариев. Подготовка программного решения к демонстрации и защите. Упаковка в JAR-файл.	3/1				Лекция-визуализация (в т.ч. в ЭИОС) Тестирование по темам лекционных занятий
	Лабораторная работа 12. Интеграция модулей. Выполнение набора тестов. Исправление дефектов.		2/1			Отчет по лабораторной работе
	Практическая работа 12. Рецензирование кода и спецификации. Демонстрация выполнения требований.			3/1		Отчёт по практической работе
	Самостоятельная работа.				11/20	Самостоятельное изучение учебных материалов
	Выполнение курсового проекта /курсовой работы				36/36	Самостоятельное выполнение с консультацией преподавателя

		18/6	**16/6**	**18/6**	**65/117**	
ИТОГО		**34/10**	**28/8**	**34/10**	**129/211**	

Примечание:-/- объем часов соответственно для очной, очно-заочной форм обучения

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

4.1. Общие методические рекомендации по освоению дисциплины, образовательные технологии

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

При проведении учебных занятий по дисциплине обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплины в форме курса, составленного на основе результатов научных исследований, проводимых университетом, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

Преподавание дисциплины ведется с применением следующих видов **образовательных технологий:**

- балльно-рейтинговая технология оценивания;
- электронное обучение;

Для оценки знаний, умений, навыков и уровня сформированности компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости студентов. В основу балльно-рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего контроля успеваемости. Максимальное количество баллов в семестре – 100.

4.2. Методические указания для обучающихся по освоению дисциплины на занятиях лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины. Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала.

Лекционные занятия проводятся в поточной аудитории с применением мультимедийного проектора в виде учебной презентации или в ЭИОС университета.

В ходе лекционных занятий рекомендуется конспектирование учебного материала.

Возможно ведение конспекта лекций в виде интеллект-карт.

Отдельные темы предлагаются для самостоятельного изучения (конспектируются).

Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям / лабораторным работам и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

4.3. Методические указания для обучающихся по освоению дисциплины на лабораторных работах

Подготовку к каждой лабораторной работе студент должен начать с ознакомления с планом занятия, который отражает содержание предложенной темы. Каждая выполненная работа с оформленным отчетом по ней подлежит защите преподавателю.

При оценивании лабораторных работ учитывается следующее:

- качество выполнения экспериментально-практической части работы и степень соответствия результатов работы заданным требованиям;

- качество оформления отчета по работе;

- качество устных ответов на контрольные вопросы при защите работы.

Лабораторные работы организуются, в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

4.4. Методические указания для обучающихся по освоению дисциплины на занятиях семинарского типа/ на практических занятиях

Практические (семинарские) занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы. Основной формой проведения семинаров и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также решение задач и разбор примеров и ситуаций в аудиторных условиях.

Практические (семинарские) занятия обучающихся обеспечивают:

- проверку и уточнение знаний, полученных на лекциях;

- получение умений и навыков составления докладов и сообщений, обсуждения вопросов по учебному материалу дисциплины;

- подведение итогов занятий по рейтинговой системе, согласно технологической карте дисциплины.

Практические занятия организуются, в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

4.5. Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающегося к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающегося на занятиях и в качестве выполненных практических заданий и других форм текущего контроля.

Самостоятельная работа студентов включает:

1. Изучение учебной литературы по курсу.

2. Работу с ресурсами Интернет

3. Самостоятельное изучение учебных материалов

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы.

Для обучающихся по заочной форме обучения самостоятельная работа является основным видом учебной деятельности.

Для обеспечения самостоятельной работы обучающихся используется электронный учебный курс, созданный в ЭИОС университета http://sdo.tolgas.ru/.

4.6. Методические указания для выполнения курсового проекта / работы

Выполнение курсового проекта/ работы способствует лучшему освоению обучающимися учебного материала, формирует практический опыт и умения по изучаемой дисциплине, способствует формированию у обучающихся готовности к самостоятельной профессиональной деятельности, является этапом к выполнению выпускной квалификационной работы.

Примерная тематика курсовых работ (проектов)

1. Архитектура Java: JVM, JRE, JDK. Компиляция и интерпретация байт-кода.
2. Обзор сред разработки (IDE). Структура Java-программы: классы, метод main().
3. Базовые примитивные типы данных в Java. Ссылочные типы данных. Переменные и константы.
4. Операции в Java: арифметические, сравнения, логические, побитовые, тернарный оператор.
5. Преобразование и приведение типов данных в Java.
6. Ввод/вывод данных: класс Scanner, System.out.println(). Форматированный вывод.
7. Логический тип boolean. Операции сравнения и логические операторы (&&, ||, !).
8. Условный оператор if, if-else. Множественный выбор с использованием switch.
9. Вложенные ветвления. Проверка предусловий. Обработка некорректных входных данных.
10. Использование метода equals() для сравнения строк.
11. Циклы в Java: for, while, do-while. Управление циклом (операторы break, continue).

12. Счетчики и накопители в циклах. Организация накопительных вычислений.
13. Вложенные циклы. Генерация числовых последовательностей.
14. Циклы для обработки строк. Бесконечные циклы и контроль завершения. Итераторы.
15. Массивы в Java: одномерные и многомерные. Объявление, инициализация, заполнение.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

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

Вся литература, включенная в данный перечень, представлена в виде электронных ресурсов в электронной библиотеке университета (ЭБС). Литература, используемая в печатном виде, представлена в научной библиотеке университета в объеме не менее 0,25 экземпляров на одного обучающегося.

Основная литература

1. Советов, Б. Я. Информационные технологии: теоретические основы : учеб. пособие для студентов вузов по направлению подгот. бакалавра «Информац. системы и технологии» / Б. Я. Советов, В. В. Цехановский. – 3-е изд., стер. – Санкт-Петербург [и др.] : Лань, 2022. – 442 с. – URL: https://reader.lanbook.com/book/264935. – ISBN 978-5-507-45305-4.
2. Сьерра, К. Изучаем Java на примерах и задачах / К. Сьерра, Б. Бейтс. – 2-е изд. – СПб. : Питер, 2024. – 720 с. – ISBN 978-5-4461-2049-3.
3. Хорстманн, К. Java. Библиотека профессионала. Том 1. Основы / К. Хорстманн. – 12-е изд. – М. : Диалектика, 2024. – 928 с. – ISBN 978-5-907515-71-2.
4. Шилдт, Г. Java. Полное руководство / Г. Шилдт. – 12-е изд. – М. : Диалектика, 2024. – 1488 с. – ISBN 978-5-907515-31-6.

Дополнительная литература

5. Остроух, А. В. Системы искусственного интеллекта : монография / А. В. Остроух, Н. Е. Суркова. – 4-е изд., стер. – Санкт-Петербург [и др.] : Лань, 2024. – 228 с. – URL: https://reader.lanbook.com/book/379988. – ISBN 978-5-507-47478-3.
6. Цифровая экономика : учеб. / И. А. Хасаншин, А. А. Кудряшов, Е. В. Кузьмин, А. А. Крюкова ; под ред. И. А. Хасаншина. – Москва : Горячая линия-Телеком, 2021. – 287 с. : ил. – ISBN 978-5-9912-0791-1.
7. Урма, Р. Современный Java: рецепты функционального программирования / Р. Урма, М. Фуско, А. Майкрофт. – М. : ДМК Пресс, 2023. – 412 с. – ISBN 978-5-93700-201-7.
8. Эванс, Э. Предметно-ориентированное проектирование (DDD) / Э. Эванс. – М. : Вильямс, 2023. – 448 с. – ISBN 978-5-8459-2045-0. – Текст : непосредственный.
9. Уорбэртон, Р. Java 17: Функциональное программирование и Stream API / Р. Уорбэртон. – СПб. : БХВ-Петербург, 2023. – 304 с. – ISBN 978-5-9775-6880-7.

5.2. Профессиональные базы данных, информационно-справочные системы, интернет-ресурсы

1. eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . - URL: https://elibrary.ru (дата обращения: 14.06.2022). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.
2. КонсультантПлюс : справочная правовая система : сайт / ЗАО «КонсультантПлюс». – Москва, 1992 - . - URL: http://www.consultant.ru (дата обращения 14.06.2022). - Текст : электронный.
3. Электронная библиотечная система Поволжского государственного университета сервиса : сайт / ФГБОУ ВО «ПВГУС». – Тольятти, 2010 - . - URL. : http://elib.tolgas.ru (дата обращения 14.06.2022). - Режим доступа: для авториз. пользователей. - Текст : электронный.
4. Электронно-библиотечная система Znanium.com : сайт / ООО "ЗНАНИУМ". – Москва, 2011 - . - URL: https://znanium.com/ (дата обращения 14.06.2022). - Режим доступа: для авториз. пользователей. - Текст : электронный.
5. Электронно-библиотечная система Лань : сайт / ООО "ЭБС ЛАНЬ". - Москва, 2011 - . - URL: https://e.lanbook.com/ (дата обращения 14.06.2022). - Режим доступа: для авториз. пользователей. - Текст : электронный.

5.3. Программное обеспечение

Информационное обеспечение учебного процесса по дисциплине осуществляется с использованием следующего программного обеспечения (лицензионного и свободно распространяемого), в том числе отечественного производства:

№ п/п	Наименование	Условия доступа
1	Microsoft Windows	из внутренней сети университета (лицензионный договор)
2	Microsoft Office	из внутренней сети университета (лицензионный договор)
3	КонсультантПлюс	из внутренней сети университета (лицензионный договор)
4	СДО MOODLE	из любой точки, в которой имеется доступ к сети Интернет (лицензионный договор)
5	Google	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое)
6	Yandex	из любой точки, в которой имеется доступ к сети Интернет (свободно распространяемое)
7	Среда разработки и компилятор/интерпретатор языка программирования, используемого на занятиях	свободно распространяемое или установленное в компьютерном классе программное обеспечение; конкретный язык определяется учебным заданием

6. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных учебным планом и рабочей программой дисциплины, оснащенные оборудованием и техническими средствами обучения.

Занятия лекционного типа. Учебные аудитории для занятий лекционного типа укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации (стационарные или переносные наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук), учебно-наглядные пособия (презентации по темам лекций), обеспечивающие тематические иллюстрации, соответствующие данной программе дисциплины.

Занятия семинарского типа. Учебные аудитории для занятий семинарского типа укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации (стационарные или переносные наборы демонстрационного оборудования (проектор, экран, компьютер/ноутбук). Для проведения лабораторных работ используется учебная аудитория «Лаборатория информационных технологий, информатики и методов программирования // Учебная аудитория для проведения занятий лекционного типа, семинарского типа, курсового проектирования, консультаций, текущего контроля и промежуточной аттестации // Помещение для хранения и профилактического обслуживания учебного оборудования (Г-413)», оснащенная следующим оборудованием: рабочие места обучающихся; Рабочее место преподавателя; Учебно-наглядные пособия, обеспечивающие тематические иллюстрации. Компьютер Асус Intel G530\iH61\GeForceGT430\4096 Mb DDR3\500Gb\LCD 22" - 14 шт., Компьютер в сборе iP2.0 LCD - 1 шт., принтер HP 1200 - 1 шт., мобильный экран проекционный DraperLuma - 1 шт., мобильный проектор Sanyo PLC, Сетевое оборудование D-Link DES-1024D, 24-портовый; локальная сеть 100 мб с доступом в Интернет 50 мб. Лаборатория оснащена кондиционером.

Лабораторные работы. Для проведения лабораторных работ используется учебная аудитория «Лаборатория информационных технологий, информатики и методов программирования // Учебная аудитория для проведения занятий лекционного типа, семинарского типа, курсового проектирования, консультаций, текущего контроля и промежуточной аттестации // Помещение для хранения и профилактического обслуживания учебного оборудования (Г-413)», оснащенная следующим оборудованием: рабочие места обучающихся; Рабочее место преподавателя; Учебно-наглядные пособия, обеспечивающие тематические иллюстрации. Компьютер Аксус Intel G530\iH61\GeForceGT430\4096 Mb DDR3\500Gb\LCD 22" - 14 шт., Компьютер в сборе iP2.0 LCD - 1 шт., принтер HP 1200 - 1 шт., мобильный экран проекционный DraperLuma - 1 шт., мобильный проектор Sanyo PLC, Сетевое оборудование D-Link DES-1024D, 24-портовый; локальная сеть 100 мб с доступом в Интернет 50 мб. Лаборатория оснащена кондиционером.

Промежуточная аттестация. Для проведения промежуточной аттестации по дисциплине используются компьютерные классы, оснащенные компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета и/или учебные аудитории, укомплектованные мебелью и техническими средствами обучения.

Самостоятельная работа. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде университета. Для организации самостоятельной работы обучающихся используются:

- компьютерные классы университета;

- библиотека (медиазал), имеющая места для обучающихся, оснащенные компьютерами с доступом к базам данных и сети Интернет.

Электронная информационно-образовательная среда университета (ЭИОС). Каждый обучающийся в течение всего периода обучения обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде университета (ЭИОС) http://sdo.tolgas.ru/ из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети "Интернет", как на территории университета, так и вне ее.

ЭИОС университета обеспечивает:

- доступ к учебным планам, рабочим программам дисциплин (модулей), программам практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), программах практик;

- формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы.

В случае реализации образовательной программы с применением электронного обучения, дистанционных образовательных технологий ЭИОС дополнительно обеспечивает:

фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательной программы;

проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети "Интернет".

7. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБУЧЕНИЯ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

При необходимости рабочая программа дисциплины может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе для дистанционного обучения. Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии (ПМПК).

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида, могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (ФОНД ОЦЕНОЧНЫХ СРЕДСТВ) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Для оценки знаний, умений, навыков и уровня сформированности компетенции по дисциплине применяется балльно-рейтинговая система контроля и оценки успеваемости студентов. В основу балльно-рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего контроля успеваемости. Максимальное количество баллов в семестре – 100.

Шкала оценки результатов освоения дисциплины, сформированности результатов обучения

Форма промежуточной аттестации	Уровень сформированности	100-балльная шкала, %	Шкала освоения, %	5-балльная дифференцированная оценка/балл	Недифференцированная оценка
зачет / экзамен/защита КР	допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	не зачтено
	пороговый	61–85,9	61–69,9	«удовлетворительно» / 3	зачтено
	пороговый	61–85,9	70–85,9	«хорошо» / 4	зачтено
	повышенный	86–100	86–100	«отлично» / 5	зачтено

По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации в соответствии за набранными за семестр баллами (по накопительному рейтингу). Студентам, набравшим в ходе текущего контроля успеваемости по дисциплине от 61 до 100 баллов и выполнившим все обязательные виды запланированных учебных занятий, по решению преподавателя без прохождения промежуточной аттестации выставляется оценка в соответствии со шкалой оценки результатов освоения дисциплины.

Результат обучения считается сформированным (повышенный уровень), если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует повышенному уровню сформированности результатов обучения.

Результат обучения считается сформированным (пороговый уровень), если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует пороговому уровню сформированности результатов обучения.

Результат обучения считается несформированным, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует допороговому уровню.

Формы текущего контроля успеваемости

Формы текущего контроля	Количество контрольных точек	Количество баллов за 1 контр. точку	Макс. возм. кол-во баллов
Посещение лекционного занятия (конспект лекций)	6	2	12
Выполнение лабораторных работ	6	6	36
Выполнение практических работ	6	6	36
Творческий рейтинг (участие в конференциях, олимпиадах, проектной деятельности и т.п.)	1	16	16
Итого по дисциплине			**100 баллов**

Система оценивания представлена в электронном учебном курсе по дисциплине http://sdo.tolgas.ru/.

8.2. Типовые контрольные задания или иные материалы для ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

8.2.1. Типовые задания к практическим (семинарским) занятиям

Практическая работа 1. Построение блок-схем алгоритмов. Разработка программ для решения задач с использованием арифметических операций и класса Math.
Задание 1. Построение блок-схем алгоритмов по словесному описанию задачи.
Задание 2. Разработка программ для решения задач с использованием арифметических операций (+, -, *, /, %).
Задание 3. Вычисление значений математических выражений с использованием класса Math (pow(), sqrt(), abs() и др.).
Задание 4. Анализ и преобразование типов данных (приведение типов) в выражениях.

Практическая работа 2. Проектирование и проверка ветвящихся алгоритмов.
Задание 1. Проектирование и проверка ветвящихся алгоритмов с использованием оператора if-else.
Задание 2. Реализация множественного выбора с помощью оператора switch-case.
Задание 3. Составление сложных логических выражений с использованием операций сравнения и логических операторов (&&, ||, !).
Задание 4. Подбор тестовых наборов данных для проверки всех ветвей условного оператора (включая граничные случаи).

Практическая работа 3. Трассировка циклов. Решение задач на накопление суммы и произведения.
Задание 1. Трассировка (ручной подсчет) циклов с разным количеством итераций.
Задание 2. Решение задач на накопление суммы и произведения элементов последовательности.
Задание 3. Преобразование одного вида цикла (for, while, do-while) в другой.
Задание 4. Оценка числа итераций и анализ эффективности циклических алгоритмов.

Практическая работа 4. Проектирование алгоритмов обработки массивов и строк.
Задание 1. Проектирование алгоритмов обработки одномерных массивов (поиск, сортировка, замена).
Задание 2. Проектирование алгоритмов обработки двумерных массивов (матриц).
Задание 3. Анализ корректности работы с индексами массивов (выход за границы).
Задание 4. Разработка алгоритмов базовой обработки строк (поиск подстроки, замена, сравнение).

Практическая работа 5. Декомпозиция задачи на методы. Проектирование интерфейсов методов.
Задание 1. Декомпозиция (разбиение) задачи на набор методов (функций).
Задание 2. Проектирование интерфейсов программных модулей (название, входные и выходные параметры).
Задание 3. Реализация перегруженных методов для решения однотипных задач с разными типами данных.
Задание 4. Определение области видимости переменных внутри методов и программы в целом.

Практическая работа 6. Разработка тестовых наборов для программного модуля. Анализ дефектов и их устранение с использованием отладчика.
Задание 1. Разработка тестовых наборов для проверки программного модуля.
Задание 2. Анализ дефектов в программном коде и их устранение с использованием отладчика.
Задание 3. Проектирование структуры для хранения данных из файла.
Задание 4. Разработка алгоритмов с использованием конструкции try-catch для обработки ошибок ввода-вывода.

Практическая работа 7. Построение рекурсивных алгоритмов и оценка их ограничений.
Задание 1. Построение рекурсивных алгоритмов для решения задач (факториал, числа Фибоначчи, обход дерева).
Задание 2. Оценка ограничений рекурсивного подхода (глубина рекурсии, стек вызовов).
Задание 3. Сравнение рекурсивного и итеративного решения одной задачи.
Задание 4. Анализ вычислительной сложности (О-нотация) простых алгоритмов.

Практическая работа 8. Выделение объектов предметной области. Проектирование ответственности классов.
Задание 1. Выделение объектов из описания предметной области.
Задание 2. Проектирование ответственности классов (какие данные хранит и какие операции выполняет класс).
Задание 3. Разработка диаграммы классов для системы из 3-4 сущностей.
Задание 4. Реализация простого класса с полями, методами и конструкторами.

Практическая работа 9. Анализ проектных решений с использованием наследования и интерфейсов. Проектирование системы с использованием абстрактных классов и интерфейсов.
Задание 1. Анализ проектных решений с использованием наследования.
Задание 2. Устранение избыточной связанности классов с помощью интерфейсов.
Задание 3. Проектирование системы с использованием абстрактных классов и интерфейсов.
Задание 4. Демонстрация полиморфного поведения объектов в иерархии классов.

Практическая работа 10. Выбор оптимальной структуры данных для заданных операций.
Задание 1. Выбор оптимальной структуры данных (из Java Collections Framework) для заданных операций (поиск, вставка, удаление).
Задание 2. Разработка цепочек преобразования данных с использованием Stream API.
Задание 3. Сравнение производительности различных коллекций (ArrayList, LinkedList, HashSet) на типовых задачах.
Задание 4. Написание параметризованных (обобщенных) методов для работы с коллекциями разных типов.

Практическая работа 11. Анализ неоднозначных и противоречивых требований. Формирование критериев приемки и их связь с тест-кейсами.
Задание 1. Анализ неоднозначных и противоречивых требований к программе.
Задание 2. Формирование критериев приемки для функциональных требований.
Задание 3. Установление связей между критериями приемки и тестовыми сценариями (тест-кейсами).
Задание 4. Составление фрагмента технической спецификации на программный модуль.

Практическая работа 12. Рецензирование кода и спецификации. Демонстрация выполнения требований.
Задание 1. Рецензирование (peer-review) исходного кода и технической спецификации.
Задание 2. Демонстрация выполнения функциональных требований программой.
Задание 3. Подготовка краткого руководства пользователя.
Задание 4. Упаковка готового приложения в исполняемый JAR-файл.

8.2.2. Типовые задания для лабораторных работ

Лабораторная работа 1. Установка и настройка. Создание первого Java-проекта. Реализация линейных алгоритмов и вычисление арифметических выражений с использованием примитивных типов.

Задание 1. Установка и настройка среды разработки (IntelliJ IDEA / Eclipse). Создание первого Java-проекта.

Задание 2. Реализация линейных алгоритмов: вычисление арифметических выражений с использованием примитивных типов данных.

Задание 3. Организация ввода данных с клавиатуры (класс Scanner) и вывода на экран (System.out.println()).

Задание 4. Вычисление значения выражения с использованием различных математических функций.

Лабораторная работа 2. Реализация программ с условными операторами и оператором switch.

Задание 1. Реализация программы с условным оператором if-else if-else для выбора одного из нескольких вариантов.

Задание 2. Реализация программы с оператором множественного выбора switch-case.

Задание 3. Обработка некорректных входных данных с помощью условных проверок.

Задание 4. Сравнение строк с использованием метода equals().

Лабораторная работа 3. Программирование циклических вычислений. Табулирование функций с использованием циклов.

Задание 1. Программирование циклических вычислений с использованием цикла for.

Задание 2. Табулирование функции (вычисление ее значений) на заданном интервале с заданным шагом.

Задание 3. Вычисление суммы или произведения элементов последовательности с использованием цикла while.

Задание 4. Обработка строк в цикле: проверка символов, формирование новой строки.

Лабораторная работа 4. Обработка массивов и матриц. Реализация алгоритмов сортировки. Обработка строк.

Задание 1. Обработка одномерных массивов: поиск элемента, замена элементов, сортировка массива.

Задание 2. Обработка двумерных массивов (матриц): нахождение суммы элементов, главной диагонали.

Задание 3. Реализация простых алгоритмов сортировки (пузырьком, выбором).

Задание 4. Обработка строк с использованием класса StringBuilder для эффективной конкатенации.

Лабораторная работа 5. Разработка набора методов для решения предметной задачи. Объединение методов в пакеты. Вызов методов из главного класса.

Задание 1. Разработка набора методов (функций) для решения конкретной предметной задачи.

Задание 2. Организация методов в пакеты (папки) по их функциональному назначению.

Задание 3. Вызов разработанных методов из главного класса Main с передачей параметров и получением результата.

Задание 4. Демонстрация перегрузки методов.

Лабораторная работа 6. Обработка структурированных данных из текстового файла. Реализация записи и чтения данных.

Задание 1. Чтение структурированных данных (например, через запятую) из текстового файла.

Задание 2. Запись данных в текстовый файл в заданном формате.

Задание 3. Обработка исключительных ситуаций при работе с файлами (например, файл не найден).
Задание 4. Использование отладчика IntelliJ IDEA: установка точек останова, пошаговое выполнение, просмотр переменных.

Лабораторная работа 7. Реализация и сравнение рекурсивных и итеративных алгоритмов для одной и той же задачи. Разработка базовых операций с односвязным списком.
Задание 1. Реализация рекурсивного алгоритма (например, вычисление факториала или чисел Фибоначчи).
Задание 2. Реализация итеративного алгоритма для решения той же задачи.
Задание 3. Сравнение производительности рекурсивного и итеративного подходов.
Задание 4. Разработка базовых операций для работы с односвязным списком (добавление, удаление, поиск).

Лабораторная работа 8. Разработка классов предметной области. Реализация методов обработки и вывода информации.
Задание 1. Разработка классов, описывающих сущности из предметной области.
Задание 2. Реализация конструкторов, геттеров и сеттеров для этих классов.
Задание 3. Реализация методов для обработки и вывода информации о состоянии объектов.
Задание 4. Создание объектов и демонстрация их работы в главном классе.

Лабораторная работа 9. Реализация иерархии классов. Полиморфная обработка коллекции объектов. Реализация интерфейсов Comparable и Comparator для сортировки.
Задание 1. Реализация иерархии классов (базовый класс и несколько наследников).
Задание 2. Полиморфная обработка коллекции объектов разных классов из одной иерархии.
Задание 3. Реализация интерфейса Comparable для сортировки объектов по заданному полю.
Задание 4. Реализация интерфейса Comparator для создания собственного алгоритма сравнения.

Лабораторная работа 10. Обработка коллекций объектов с использованием типовых алгоритмов: поиск, сортировка, фильтрация, группировка.
Задание 1. Обработка коллекций объектов с использованием типовых алгоритмов (поиск, сортировка).
Задание 2. Фильтрация коллекции по заданному условию с использованием Stream API.
Задание 3. Группировка элементов коллекции по определенному признаку.
Задание 4. Измерение времени выполнения операций для разных коллекций (ArrayList vs LinkedList).

Лабораторная работа 11. Преобразование набора требований в техническую спецификацию для Java-компонента.
Задание 1. Преобразование набора требований (из техзадания) в техническую спецификацию для программного компонента.
Задание 2. Описание назначения, интерфейсов и структур данных для компонента.
Задание 3. Описание ограничений и критериев приемки компонента.
Задание 4. Установление связей (трассировка) между требованиями, спецификацией и кодом.

Лабораторная работа 12. Интеграция модулей. Выполнение набора тестов. Исправление дефектов.
Задание 1. Интеграция всех ранее разработанных модулей в единое приложение.
Задание 2. Выполнение набора тестов, исправление обнаруженных дефектов.
Задание 3. Документирование кода в формате Javadoc.
Задание 4. Упаковка готового приложения в исполняемый JAR-файл и подготовка к демонстрации.

8.3. Типовые контрольные задания или иные материалы для проведения ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Форма проведения промежуточной аттестации по дисциплине: *зачет* и *экзамен/ защита курсового проекта (курсовой работы) (по результатам накопительного рейтинга или в форме компьютерного тестирования).*

Устно-письменная форма по экзаменационным билетам предполагается, как правило, для сдачи академической задолженности.

Защита курсового проекта/ работы. *Результаты защиты курсового проекта/ работы выставляются по пятибалльной системе оценивания ("отлично", "хорошо", "удовлетворительно", "неудовлетворительно") с обязательным проставлением количества баллов, набранных в соответствии с балльно-рейтинговой системой (по стобалльной шкале).*

Перечень вопросов к защите курсовой работы

1. Архитектура Java: JVM, JRE, JDK. Компиляция и интерпретация байт-кода.
2. Обзор сред разработки (IDE). Структура Java-программы: классы, метод main().
3. Базовые примитивные типы данных в Java. Ссылочные типы данных. Переменные и константы.
4. Операции в Java: арифметические, сравнения, логические, побитовые, тернарный оператор.
5. Преобразование и приведение типов данных в Java.
6. Ввод/вывод данных: класс Scanner, System.out.println(). Форматированный вывод.
7. Логический тип boolean. Операции сравнения и логические операторы (&&, ||, !).
8. Условный оператор if, if-else. Множественный выбор с использованием switch.
9. Вложенные ветвления. Проверка предусловий. Обработка некорректных входных данных.
10. Использование метода equals() для сравнения строк.
11. Циклы в Java: for, while, do-while. Управление циклом (операторы break, continue).
12. Счетчики и накопители в циклах. Организация накопительных вычислений.
13. Вложенные циклы. Генерация числовых последовательностей.
14. Циклы для обработки строк. Бесконечные циклы и контроль завершения. Итераторы.
15. Массивы в Java: одномерные и многомерные. Объявление, инициализация, заполнение.

Перечень вопросов и заданий для подготовки к зачету (3 семестр)

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

1. Архитектура Java: JVM, JRE, JDK. Компиляция и интерпретация байт-кода.
2. Обзор сред разработки (IDE). Структура Java-программы: классы, метод main().
3. Базовые примитивные типы данных в Java. Ссылочные типы данных. Переменные и константы.
4. Операции в Java: арифметические, сравнения, логические, побитовые, тернарный оператор.
5. Преобразование и приведение типов данных в Java.
6. Ввод/вывод данных: класс Scanner, System.out.println(). Форматированный вывод.
7. Логический тип boolean. Операции сравнения и логические операторы (&&, ||, !).
8. Условный оператор if, if-else. Множественный выбор с использованием switch.
9. Вложенные ветвления. Проверка предусловий. Обработка некорректных входных данных.
10. Использование метода equals() для сравнения строк.
11. Циклы в Java: for, while, do-while. Управление циклом (операторы break, continue).
12. Счетчики и накопители в циклах. Организация накопительных вычислений.
13. Вложенные циклы. Генерация числовых последовательностей.
14. Циклы для обработки строк. Бесконечные циклы и контроль завершения. Итераторы.
15. Массивы в Java: одномерные и многомерные. Объявление, инициализация, заполнение.

16. Индексация и обход массивов. Алгоритмы поиска элементов в массиве.
17. Алгоритмы сортировки массивов (базовые методы).
18. Строки в Java. Основные методы работы со строками. Сравнение строк.
19. Классы StringBuilder и StringBuffer для работы с изменяемыми строками.
20. Методы (функции) в Java. Объявление методов. Передача параметров по значению.
21. Возврат значений из методов. Перегрузка методов.
22. Область видимости переменных. Статические и нестатические методы.
23. Понятие рекурсивных методов. Принцип работы рекурсии. Стек вызовов.
24. Организация пакетов и импорт классов. Модификаторы доступа (public, private, protected).
25. Принципы написания чистого кода в Java: именование, длина методов, комментарии.
26. Пользовательские типы данных в Java. Структуры данных.
27. Работа с текстовыми файлами. Работа с двоичными файлами.
28. Иерархия исключений в Java. Проверяемые и непроверяемые исключения.
29. Конструкция try-catch-finally. Ресурсный блок try-with-resources.
30. Создание собственных исключений.
31. Основы отладки в IntelliJ IDEA: точки останова, пошаговое выполнение, просмотр переменных.
32. Этапы разработки программы: постановка задачи, формализация, алгоритмизация, кодирование, тестирование.
33. Принципы тестирования программы. Тестовые случаи и регрессионная проверка.

ПК-1. Способен к выполнению работ по проектированию программного обеспечения

34. Понятие требования к программному обеспечению. Источники требований.
35. Функциональные и нефункциональные требования. Критерии качества требований.
36. Анализ реализуемости требований в контексте Java-стек технологий.
37. Декомпозиция требований на модули. Техническая спецификация программного компонента.
38. Назначение, интерфейсы, структуры данных, ограничения, критерии приемки в технической спецификации.
39. Критерии приемки и их связь с тест-кейсами.
40. Связь требований, архитектурных решений и исходного кода.

Перечень вопросов и заданий для подготовки к экзамену (4 семестр)

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

1. Архитектура Java: JVM, JRE, JDK. Компиляция и интерпретация байт-кода. Структура Java-программы.
2. Базовые примитивные типы данных. Ссылочные типы. Преобразование и приведение типов.
3. Операции в Java: арифметические, сравнения, логические, побитовые, тернарный оператор.
4. Ввод/вывод данных: класс Scanner, System.out.println(). Форматированный вывод.
5. Условный оператор if, if-else. Множественный выбор с использованием switch. Вложенные ветвления.
6. Циклы в Java: for, while, do-while. Управление циклом. Вложенные циклы.
7. Массивы в Java: одномерные и многомерные. Объявление, инициализация, индексация и обход.
8. Алгоритмы поиска и сортировки массивов.
9. Строки в Java. Основные методы. Классы String, StringBuilder, StringBuffer.
10. Методы (функции) в Java. Объявление, параметры, возврат значений. Перегрузка методов.
11. Область видимости переменных. Статические и нестатические методы. Модификаторы доступа.
12. Понятие рекурсии. Рекурсивные методы. Стек вызовов и ограничения рекурсии.

13. Сравнение итеративных и рекурсивных решений.
14. Пользовательские типы данных. Структуры данных.
15. Работа с текстовыми и двоичными файлами. Иерархия исключений в Java.
16. Конструкция try-catch-finally. try-with-resources. Создание собственных исключений.
17. Основы ООП в Java: классы и объекты. Поля, методы, конструкторы.
18. Инкапсуляция: модификаторы доступа. Геттеры и сеттеры. Жизненный цикл объекта.
19. Ключевое слово this. Композиция и агрегация. Ассоциации между классами.
20. Принципы проектирования простых классов.
21. Наследование в Java. Повторное использование кода. Переопределение методов.
22. Полиморфизм. Абстрактные классы. Интерфейсы в Java.
23. Дефолтные и статические методы в интерфейсах. Множественное наследование через интерфейсы.
24. Принцип подстановки Лисков. Композиция vs наследование.
25. Инкапсуляция, наследование и полиморфизм — три кита ООП.
26. Java Collections Framework (JCF): иерархия. Интерфейсы Collection, List, Set, Queue, Map.
27. Классы-реализации: ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap.
28. Итераторы. Цикл for-each. Обобщенные типы: параметризованные классы и методы.
29. Ограничения обобщений. Сравнение структур данных по времени выполнения операций.
30. Функциональные интерфейсы. Лямбда-выражения. Stream API: фильтрация, сортировка, группировка.
31. Основы отладки в IntelliJ IDEA. Принципы тестирования программы. Тестовые случаи.
32. Сборка программного решения из модулей. Организация зависимостей с помощью Maven/Gradle.
33. Стратегии тестирования: модульное, интеграционное, регрессионное.
34. Документирование кода (Javadoc). Документирование пользовательских сценариев.
35. Упаковка приложения в JAR-файл. Подготовка программного решения к демонстрации и защите.

ПК-1. Способен к выполнению работ по проектированию программного обеспечения

36. Понятие требования к программному обеспечению. Источники требований. Функциональные и нефункциональные требования.
37. Критерии качества требований: однозначность, проверяемость, выполнимость, непротиворечивость.
38. Анализ реализуемости требований к программному обеспечению. Выявление конфликтующих и неполных требований.
39. Декомпозиция требований на функции программной системы. Трассируемость требований и ее назначение.
40. Критерии приемки и связь с тестированием.
41. Назначение технической спецификации программного компонента. Структура технической спецификации.
42. Спецификация интерфейсов программных модулей. Спецификация структур и форматов данных.
43. Описание исключительных ситуаций и ошибок в спецификации.
44. Связь требований, архитектурных решений и исходного кода.
45. Проверка программной реализации на соответствие требованиям. Рецензирование требований и технических спецификаций.
46. Использование прототипа для уточнения требований. Документирование изменений требований.

Типовые практические задания для зачета и экзамена

1. Разработать программу обработки последовательности/массива по заданному условию (поиск, сортировка, подсчет суммы и т.д.).
2. Разработать функцию или модуль с заданным интерфейсом и тестами.

3. Спроектировать класс предметной области и реализовать проверку его инвариантов (корректности состояния).
4. Найти и исправить дефект в предложенном фрагменте программы, описать тест, выявляющий ошибку.
5. По исходному описанию сформулировать функциональные требования и фрагмент технической спецификации.
6. Проверить соответствие фрагмента реализации заданным требованиям.
7. Разработать иерархию классов и продемонстрировать полиморфное поведение.
8. Обработать коллекцию объектов с использованием Stream API (фильтрация, сортировка, группировка).