

Минобрнауки России  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)**

**УТВЕРЖДАЮ**

Заведующий кафедрой



Каменский Михаил Игоревич

Кафедра функционального анализа и операторных уравнений

25.05.23г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.ДВ.05.01 Язык программирования Java

**1. Код и наименование направления подготовки/специальности:**

10.05.04 Информационно-аналитические системы безопасности

**2. Профиль подготовки/специализация:**

Автоматизация информационно-аналитической деятельности

**3. Квалификация (степень) выпускника:**

Специалитет

**4. Форма обучения:**

Очная

**5. Кафедра, отвечающая за реализацию дисциплины:** Кафедра функционального анализа и операторных уравнений

**6. Составители программы:**

Копытин Алексей Вячеславович, доцент кафедры функционального анализа и операторных уравнений

**7. Рекомендована:**

НМС математического факультета, протокол № 0500-06 от 25.05.2023г.

**8. Учебный год:** 2027-2028

Семестр: 10

## 9. Цели и задачи учебной дисциплины:

Цели изучения дисциплины:

- получение знаний о современном объектно-ориентированном языке программирования Java и овладение основными приемами программирования, получение практических навыков разработки программ на языке Java.

Задачи учебной дисциплины:

- изучение принципов построения, проектирования и разработки компьютерных программ на языке Java
- использование современных инструментальных программных средств в профессиональной деятельности для решения конкретных задач.

## 10. Место учебной дисциплины в структуре ООП:

дисциплина Язык программирования Java относится к части, формируемой участниками образовательных отношений, блока Б1

## 11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников) и индикаторами их достижения:

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
ПК-1 Способен обеспечивать функционирование средств защиты информации в информационноаналитических системах	ПК-1.1 Владеет средствами защиты информации в ИАС	Знать: средства защиты информации в ИАС. Уметь: использовать средства защиты информации в ИАС. Владеть: средствами защиты информации в ИАС.
ПК-3 Способен решать типовые задачи обработки и анализа информации в информационноаналитических системах государственных органов, обеспечивающих национальную безопасность	ПК-3.1 Владеет способами решения типовых задач обработки и анализа информации в информационноаналитических системах	Знать: способы решения типовых задач обработки и анализа информации в информационноаналитических системах. Уметь: решать типовые задачи обработки и анализа информации в информационноаналитических системах. Владеть: способами решения типовых задач обработки и анализа информации в информационноаналитических системах.

<p>ПК-3 Способен решать типовые задачи обработки и анализа информации в информационноаналитических системах государственных органов, обеспечивающих национальную безопасность</p>	<p>ПК-3.2 Способен выбирать подходящие методы решения задач обработки информации в информационноаналитических системах</p>	<p>Знать: подходящие методы решения задач обработки информации в информационноаналитических системах. Уметь: выбирать подходящие методы решения задач обработки информации в информационноаналитических системах. Владеть: способностью выбирать подходящие методы решения задач обработки информации в информационноаналитических системах.</p>
---	--	--

**12. Объем дисциплины в зачетных единицах/час:**

5/180

**Форма промежуточной аттестации:**

Экзамен

**13. Трудоемкость по видам учебной работы**

Вид учебной работы	Семестр 10	Всего
Аудиторные занятия	96	96
Лекционные занятия	48	48
Практические занятия		0
Лабораторные занятия	48	48
Самостоятельная работа	48	48
Курсовая работа		0
Промежуточная аттестация	36	36
Часы на контроль	36	36
Всего	180	180

**13.1. Содержание дисциплины**

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
1	Лекции		
1.1	Введение в предмет	Цели и задачи изучения дисциплины; понятие и свойства алгоритма; краткий обзор языка Java; примеры и разбор простейших программ на языке Java; краткий обзор сред разработки Javaпрограмм; создание проекта в среде разработки	
1.2	Переменные и типы данных, ввод-вывод данных	Понятие переменной, понятие типа данных и строгой типизации; стандартные типы языка Java (примитивные и String), преобразования типов; особенности хранения различных типов в памяти компьютера и их обработки; System.in и System.out, ввод данных с помощью класс java.util.Scanner, форматирование вывода с помощью printf и String.format	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК

1.3	Функции	<p>Понятие функций (статических методов класса применительно к Java): описание и вызов, передача параметров; проектирование программы с использованием функций; важность разделения программы на подзадачи и превильного выделения подзадач, структуризация и принципы повторного использования кода; примеры программ с использованием функций</p>	
1.4	Операторы управления ходом выполнения программ	<p>Условный оператор, составной оператор, тернарный оператор, операторы циклов в Java, операторы break и continue; вложенные блоки кода и видимость переменных; соглашения по оформлению Java-кода; примеры решения задач</p>	
1.5	Простейшие алгоритмы	<p>Примеры решения задач: перевод десятичного числа в двоичное представление, собственная реализация sqrt методом половинного деления, вычисление числа Пи, выделение n-ой части строки, разделенной запятыми, печать n символов последовательности и др.; разные варианты решения задач и выбор оптимального варианта</p>	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
-----	---------------------------------	-------------------------------	---

1.6	Составные типы данных	<p>Массивы и множественные массивы (массивы массивов); типичные задачи обработки массивов, класс <code>java.util.Arrays</code>; разработка библиотеки функций ввода/вывода массивов и других функции в виде класс <code>ArrayUtils</code>.</p> <p>Перечисления (<code>enum</code>). Структуры данных в виде простейших классов, важность применения структур для упрощения и структуризации кода, массивы структур, примеры задач. Массивы и структуры в качестве параметров функций и возвращаемых значений.</p> <p>Понятие типов-значений (<code>value types</code>) и ссылочных типов (<code>reference types</code>) данных в Java, понятие объектов, ссылочная модель и сборка мусора.</p>	
1.7	Строки	<p>Строки в Java, особенности класса <code>String</code> особенности реализации (неизменяемость), поддерживаемые методы, сравнение строк с помощью <code>equals</code>.</p> <p>Особенности конкатенации строк и класс <code>StringBuilder</code>.</p> <p>Поверхностное знакомство с регулярными выражения и возможностями их применения (<code>RegExp</code> и некоторые методы <code>String</code>).</p>	
п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК

1.8	Типичные задачи обработки массивов и строк	<p>Примеры решения задач: поиск минимума/максимума в массиве, поиск индекса элемента, сортировка массива методом "пузырька", передача различных критериев сортировки в метод Arrays.sort, бинарный поиск в упорядоченном массиве, операции со столбцами и строка в двумерном массиве и др.; демонстрации важности структуризации программы с помощью функций на примере задач обработки массивов и строк</p>	
1.9	Основы объектноориентированного подхода	<p>Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между static функциями и методами классов; классы как расширение концепции типа данных в виде объединения данных и методов их обработки, понятие состояния объекта, примеры; принципы инкапсуляции; понятие наследования и полиморфизма; класс Object и его методы</p>	
1.10	Создание приложений с оконным интерфейсом	<p>Принципы создания приложений с оконным интерфейсом; библиотека Swing и базовые Swing-компоненты; визуальное проектирование форм (JFrame) в среде разработки, обработка событий компонентов; понятие модели данных для сложных копонентов, JTable и разработка класса JTableUtils для упрощения работы с JTable. Типичная архитектура приложений с оконным интерфейсом и важность разделения логики и отображения, приложения с несколькими формами, примеры приложений. Второй вариант построения оконного интерфейса - с помощью JavaFX, возможности и особенности, примеры приложений.</p>	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
1.11	Коллекции	<p>Понятие коллекций, какие виды коллекций бывают (списки, словари, множества, стеки и очереди), иерархия коллекций в Java; списки (List&lt;T&gt;) по сравнению с массивами, методы списков; словари (Map&lt;K, V&gt;) и множества (Set&lt;T&gt;) и их методы; понятие обобщенных типов данных и кода (generics); разница между интерфейсом и реализацией (List&lt;T&gt; и ArrayList&lt;T&gt;/LinkedList&lt;T&gt;, Map&lt;K, V&gt; и TreeMap&lt;K, V&gt;/HashMap&lt;K, V&gt;); примеры эффективного решения задач с помощью коллекций</p>	
1.12	Создание прикладных приложений (создание игры)	<p>Пример создания полноценного оконного приложения на Java - игры "Сапер" в качестве демонстрации применения структур данных (классов, массивов) и алгоритмов применительно к конкретной практической задаче; архитектура приложения с применением ООП подхода выделение логики в виде отдельного класса, реализация отображения в JTable; знакомство с классом java.awt.Graphics</p>	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК

1.13	Рекурсия и рекурсивные алгоритмы	<p>Понятие рекурсии в программировании; прямая и косвенная рекурсия; вычисление факториала, когда стоит и когда не стоит использовать рекурсию; рекурсивное вычисление чисел Фибоначчи - экспоненциальный рост количества повторных вызовов функций в некоторых рекурсивных алгоритмах и возможное решение с помощью кеширования результатов вычислений; примеры решения задач: рисование треугольника Серпинского, сопоставление строки шаблону, обход двумерного поля в глубину и ширину; варианты реализации истинно рекурсивных алгоритмов без рекурсивных вызовов с применением стеков (Stack&lt;T&gt;) и очередей (Queue&lt;T&gt;)</p>	
2	Практические занятия		
2.1	нет		
3	Лабораторные работы		
3.1	Переменные и типы данных, ввод-вывод данных	<p>Понятие переменной, понятие типа данных и строгой типизации; стандартные типы языка Java (примитивные и String), преобразования типов; особенности хранения различных типов в памяти компьютера и их обработки; System.in и System.out, ввод данных с помощью класса java.util.Scanner, форматирование вывода с помощью printf и String.format</p>	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
3.2	Функции	<p>Понятие функций (статических методов класса применительно к Java): описание и вызов, передача параметров; проектирование программы с использованием функций; важность разделения программы на подзадачи и превильного выделения подзадач, структуризация и принципы повторного использования кода; примеры программ с использованием функций</p>	
3.3	Операторы управления ходом выполнения программ	<p>Условный оператор, составной оператор, тернарный оператор, операторы циклов в Java, операторы break и continue; вложенные блоки кода и видимость переменных; соглашения по оформлению Java-кода; примеры решения задач</p>	
3.4	Простейшие алгоритмы	<p>Примеры решения задач: перевод десятичного числа в двоичное представление, собственная реализация sqrt методом половинного деления, вычисление числа Пи, выделение n-ой части строки, разделенной запятыми, печать n символов последовательности и др.; разные варианты решения задач и выбор оптимального варианта</p>	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
3.5	Составные типы данных	<p>Массивы и множественные массивы (массивы массивов); типичные задачи обработки массивов, класс <code>java.util.Arrays</code>; разработка библиотеки функций ввода/вывода массивов и других функции в виде класс <code>ArrayUtils</code>.</p> <p>Перечисления (<code>enum</code>). Структуры данных в виде простейших классов, важность применения структур для упрощения и структуризации кода, массивы структур, примеры задач.</p> <p>Массивы и структуры в качестве параметров функций и возвращаемых значений.</p> <p>Понятие типов-значений (<code>value types</code>) и ссылочных типов (<code>reference types</code>) данных в Java, понятие объектов, ссылочная модель и сборка мусора.</p>	
3.6	Строки	<p>Строки в Java, особенности класса <code>String</code> особенности реализации (неизменяемость), поддерживаемые методы, сравнение строк с помощью <code>equals</code>.</p> <p>Особенности конкатенации строк и класс <code>StringBuilder</code>.</p> <p>Поверхностное знакомство с регулярными выражения и возможностями их применения (<code>RegExp</code> и некоторые методы <code>String</code>).</p>	

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
3.7	Типичные задачи обработки массивов и строк	<p>Примеры решения задач: поиск минимума/максимума в массиве, поиск индекса элемента, сортировка массива методом "пузырька", передача различных критериев сортировки в метод Arrays.sort, бинарный поиск в упорядоченном массиве, операции со столбцами и строка в двумерном массиве и др.; демонстрации важности структуризации программы с помощью функций на примере задач обработки массивов и строк</p>	
3.8	Основы объектноориентированного подхода	<p>Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между static функциями и методами классов; классы как расширение концепции типа данных в виде объединения данных и методов их обработки, понятие состояния объекта, примеры; принципы инкапсуляции; понятие наследования и полиморфизма; класс Object и его методы</p>	

3.9	Создание приложений с оконным интерфейсом	<p>Принципы создания приложений с оконным интерфейсом; библиотека Swing и базовые Swing-компоненты; визуальное проектирование форм (JFrame) в среде разработки, обработка событий компонентов; понятие модели данных для сложных компонентов, JTable и разработка класса JTableUtils для упрощения работы с JTable.</p> <p>Типичная архитектура приложений с оконным интерфейсом и важность разделения логики и отображения, приложения с несколькими формами, примеры приложений. Второй вариант построения оконного интерфейса - с помощью JavaFX, возможности и особенности, примеры приложений.</p>	
-----	---	---	--

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
3.10	Коллекции	<p>Понятие коллекций, какие виды коллекций бывают (списки, словари, множества, стеки и очереди), иерархия коллекций в Java; списки (List&lt;T&gt;) по сравнению с массивами, методы списков; словари (Map&lt;K, V&gt;) и множества (Set&lt;T&gt;) и их методы; понятие обобщенных типов данных и кода (generics); разница между интерфейсом и реализацией (List&lt;T&gt; и ArrayList&lt;T&gt;/LinkedList&lt;T&gt;, Map&lt;K, V&gt; и TreeMap&lt;K, V&gt;/HashMap&lt;K, V&gt;); примеры эффективного решения задач с помощью коллекций</p>	

3.11	Создание прикладных приложений (создание игры)	Пример создания полноценного оконного приложения на Java - игры "Сапер" в качестве демонстрации применения структур данных (классов, массивов) и алгоритмов применительно к конкретной практической задаче; архитектура приложения с применение ООП подхода выделение логики в виде отдельного класса, реализация отображения в JTable; знакомство с классом java.awt.Graphics	
п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайнкурса, ЭУМК
3.12	Рекурсия и рекурсивные алгоритмы	Понятие рекурсии в программировании; прямая и косвенная рекурсия; вычисление факториала, когда стоит и когда не стоит использовать рекурсию; рекурсивное вычисление чисел Фибоначчи - экспоненциальный рост количества повторных вызовов функций в некоторых рекурсивных алгоритмах и возможные решения с помощью кеширования результатов вычислений; примеры решения задач: рисование треугольника Серпинского, сопоставление строки шаблону, обход двумерного поля в глубину и ширину; варианты реализации истинно рекурсивных алгоритмов без рекурсивных вызовов с применением стеков (Stack<T>) и очередей (Queue<T>)	

**13.2. Темы (разделы) дисциплины и виды занятий**

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего

1	Введение в предмет	2			2	4
2	Переменные и типы данных, ввод-вывод данных	3		4	3	10
3	Функции	4		4	4	12
4	Операторы управления ходом выполнения программ	4		4	4	12
5	Простейшие алгоритмы	3		4	3	10
6	Составные типы данных	4		4	4	12
7	Строки	4		4	4	12
8	Типичные задачи обработки массивов и строк	4		4	4	12
9	Основы объектноориентированного подхода	4		4	4	12
10	Создание приложений с оконным интерфейсом	4		4	4	12
11	Коллекции	4		4	4	12
12	Создание прикладных приложений (создание игры)	4		4	4	12
13	Рекурсия и рекурсивные алгоритмы	4		4	4	12

		48	0	48	48	144
--	--	----	---	----	----	-----

#### 14. Методические указания для обучающихся по освоению дисциплины

Рекомендуется работа с конспектами лекций, презентационным материалом, выполнение всех лабораторных и контрольных работ.

#### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

№ п/п	Источник
1	Эккель, Брюс. Философия Java = Thinking in Java / Брюс Эккель ; [пер. с англ. Е. Матвеева] .— 4-е полное изд. — Москва : Вильямс, 2017 .— 1165 с. : ил. — (Классика computer science) .— ISBN 978-5-496-01127-3.
2	Хорстманн, Кей. Java = Core Java / Кей Хорстманн ; [пер. с англ. и ред. И.В. Берштейна] .— Москва ; Санкт-Петербург ; Киев : Вильямс, 2017- .— (Библиотека профессионала) .— ISBN 978-5-8459-2083-6.
3	Блох, Джошуа. Java. Эффективное программирование = Effective Java programming language guide / Джошуа Блох ; пер. с англ. В. Стрельцов ; науч. ред. Р. Усманов ; предисл. Г. Стила .— Москва : Лори, 2017 .— 294 с. : табл. — (Серия Java "...из первых рук") .— Библиогр.: с. 288-294 .— ISBN 978-5-85582-347-9.

#### б) дополнительная литература:

№ п/п	Источник
1	Хорстманн, Кей. Java = Core Java / К. Хорстманн, Г. Корнелл ; [пер. с англ. и ред. И.В. Берштейна] .— Москва ; Санкт-Петербург ; Киев : Вильямс, 2015 .— (Библиотека профессионала) .— ISBN 978-5-8459-2032-4.
2	Шилдт, Герберт. Java : руководство для начинающих / Герберт Шилдт ; [пер. с англ. и ред. В.В. Вейтмана] .— 4-е изд. — М. [и др.] : Вильямс, 2009 .— 715 с. : ил. — Предм. указ.: с.709-715 .— ISBN 978-5-8459-1440-8.

3	Портянкин, Иван Александрович. Swing. Эффективные пользовательские интерфейсы. Java Foundation Classes / Иван Портянкин .— СПб. [и др.] : Питер, 2005 .— 523 с. — (Библиотека программиста) .— Алф. указ.: с.515-523 .— ISBN 5469-00005-2.
4	Дейтел, Х.М. Как программировать на Java / Х.М. Дейтел, П.Д. Дейтел .— М. : Бинум, 2003.Кн. 1: Основы программирования / Пер. с англ. под ред. А.В. Козлова .— 4-е изд. — 2003 .— 847 с. : ил. — Парал. тит. л. англ. — ISBN 5-9518-0015-3.
5	Технологии программирования на Java 2 / Х.М. Дейтел, П.Д. Дейтел, С.И. Сантри .— М. : Бинум, 2003-.Кн. 1: Графика, JavaBeans, интерфейс пользователя / Пер. с англ. под. ред. А.И. Тихонова .— 2003 .— 560 с. : ил. — Парал. тит. л. англ. — ISBN 5-9518-0017-X .— ISBN 0-13-089560-1.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	Самоучитель по Java с нуля [Электронный ресурс] : — Режим доступа: <a href="https://vertex-academy.com/tutorials/ru/samouchitel-po-java-s-nulya/">https://vertex-academy.com/tutorials/ru/samouchitel-po-java-s-nulya/</a>
2	Учебник: программирование на Java [Электронный ресурс] : — Режим доступа: <a href="https://java9.ru/">https://java9.ru/</a>
3	Иллюстрированный самоучитель по Java [Электронный ресурс] : — Режим доступа: <a href="http://www.realcoding.net/teach/java/">http://www.realcoding.net/teach/java/</a>

**16. Перечень учебно-методического обеспечения для самостоятельной работы**

№ п/п	Источник
1	Шилдт, Герберт. Искусство программирования на JAVA : пер. с англ. / Герберт Шилдт, Джеймс Холмс .— СПб. [и др.] : БХВ-Петербург, 2005 .— 331 с. : ил. — Парал. тит. л. англ. — Предм. указ. : с.330-331, 4000 экз.

2	Лафоре, Роберт. Структуры данных и алгоритмы в Java = Data structures @ algorithms in Java / Роберт Лафоре ; [пер. с англ. Е. Матвеева] .— 2-е изд. — Санкт-Петербург [и др.] : Питер, 2014 .— 701 с. : ил., табл. — (Классика computer science) .— Библиогр.: с.683-685 .— Алф. указ.: с.695-701 .— ISBN 985-5-49600740-5.
---	---

**17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):**

OpenJDK - бесплатен

Среда разработки NetBeans или IntelliJ IDEA (академическая лицензия или версия Community) бесплатны

**18. Материально-техническое обеспечение дисциплины:**

Специализированная мебель, маркерная доска, персональные компьютеры

**19. Оценочные средства для проведения текущей и промежуточной аттестаций**

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
1	1. Введение в предмет 2. Переменные и типы данных, ввод-вывод данных 3. Функции 4. Операторы управления ходом выполнения программ 5. Простейшие алгоритмы 6. Составные типы данных 7. Строки 8. Типичные задачи обработки массивов и строк 9. Основы объектноориентированного подхода 10. Создание приложений с оконным интерфейсом 11. Коллекции 12. Создание прикладных приложений (создание игры) 13. Рекурсия и рекурсивные алгоритмы	ПК-1	ПК-1.1	Обязательные практические задания (контроль и оценка выполнения)

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
2	<p>1. Введение в предмет</p> <p>2. Переменные и типы данных, ввод-вывод данных</p> <p>3. Функции</p> <p>4. Операторы управления ходом выполнения программ</p> <p>5. Простейшие алгоритмы</p> <p>6. Составные типы данных</p> <p>7. Строки</p> <p>8. Типичные задачи обработки массивов и строк</p> <p>9. Основы объектноориентированного подхода</p> <p>10. Создание приложений с оконным интерфейсом</p> <p>11. Коллекции</p> <p>12. Создание прикладных приложений (создание игры)</p> <p>13. Рекурсия и рекурсивные алгоритмы</p>	ПК-3	ПК-3.1	Обязательные практические задания (контроль и оценка выполнения)

3	1. Введение в предмет 2. Переменные и типы данных, ввод-вывод данных 3. Функции 4. Операторы управления ходом выполнения программ 5. Простейшие алгоритмы 6. Составные типы данных 7. Строки 8. Типичные задачи обработки массивов и строк 9. Основы объектноориентированного подхода 10. Создание приложений с оконным интерфейс 11. Коллекции 12. Создание прикладных приложений (создание игры) 13. Рекурсия и рекурсивные алгоритмы	ПК-3	ПК-3.2	Обязательные практические задания (контроль и оценка выполнения)
---	---	------	--------	--

Промежуточная аттестация Форма контроля - Экзамен  
Оценочные средства для промежуточной аттестации

Для оценивания результатов обучения на экзамене используются следующие содержательные показатели (формулируется с учетом конкретных требований дисциплины):

- 1) знание теоретических основ учебного материала, основных определений, понятий используемой терминологии;
- 2) умение проводить обоснование и представление основных теоретических и практических результатов (теорем, алгоритмов, методик) с использованием математических выкладок, блоксхем, структурных схем и стандартных описаний к ним;
- 3) умение связывать теорию с практикой, иллюстрировать ответ примерами, в том числе, собственными, умение выявлять и анализировать основные закономерности, полученные, в том числе, в ходе выполнения лабораторно-практических заданий;
- 4) умение обосновывать свои суждения и профессиональную позицию по излагаемому вопросу;
- 5) владение навыками программирования и экспериментирования в рамках выполняемых лабораторных заданий;

Различные комбинации перечисленных показателей определяют критерии оценивания результатов обучения (сформированности компетенций) на зачете:

- высокий (углубленный) уровень сформированности компетенций;
- повышенный (продвинутый) уровень сформированности компетенций;
- пороговый (базовый) уровень сформированности компетенций.

Для оценивания результатов обучения на зачете с оценкой используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

## **20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания**

### **20.1 Текущий контроль успеваемости**

Задача 1 - Запись выражений и оператор присвоения

Задача 2 - Условный оператор

Задача 3 - Применение функций

Задача 4 - Циклы

Задача 5 - Одномерные массивы

Задача 6 - Двумерные массивы

Задача 7 - Коллекции

Задача 8 - Структуры данных

Задача 9 - Строки

Задача 10 - Рекурсия

Задача 11 - Логическая игра

## **20.2 Промежуточная аттестация**

1. Алгоритм и его свойства
2. Обзор языка Java
3. Типы данных языка Java: типы-значения и ссылочные типы, обзор числовых типов
4. Переменные, область видимости переменных, строгая типизация
5. Строки и работа со строками (обзор String, StringBuilder, RegExp)
6. Операторы языка Java
7. Соглашения по оформлению Java-кода
8. Функции, структуризация программ с помощью функций
9. Ввод-вывод данных в Java
10. Массивы: одномерные, двумерные, типичные задачи с использованием массивов
11. Сортировка: реализация пузырьковой сортировки, Arrays.sort, различные критерии сортировки
12. Поиск в массиве: последовательный поиск, бинарный в отсортированном массиве
13. Составные типы данных - классы
14. Основы объектно-ориентированного подхода
15. Коллекции в языке Java
16. Практические примеры работы со словарями (Map)
17. Обобщенное программирование (generics), классы-обертки над примитивными типами данных
18. Построение оконного интерфейса с помощью библиотеки Swing, архитектура приложения
19. Работа с компонентом JTable
20. "Рисование" в Java, обзор методов класса Graphics
21. Обзор библиотеки и принципов создания приложений JavaFX
22. Рекурсия и рекурсивные алгоритмы
23. Рекурсивные алгоритмы: сопоставление строки шаблону
24. Рекурсивные алгоритмы: рисование фракталов
25. Рекурсивные алгоритмы: обход поля в глубину
26. Реализация рекурсивных алгоритмов без рекурсии с помощью стека и очереди

## **20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ**

1) закрытые задания (тестовые, средний уровень сложности):

1. Строгая типизация предполагает (выберите верные утверждения)?
  - Все используемые в функции переменные должны объявляться строго до остального кода функции.
  - При компиляции программы весь код (все операции) проверяется на совместимость или возможность преобразования типов, несовместимость считается ошибкой.
  - В программе нельзя определить несколько функций с одинаковым именем.
  - Язык программирования обязательно должен быть объектно-ориентированным.
2. Какие утверждения верны для массивов в языке Java?
  - Размер массива может быть изменен после его создания.
  - Индексация элементов в массиве начинается с 1.
  - Все элементы в конкретном массиве должны быть одного типа (или наследоваться от одного типа).
  - В одной программе могут использоваться массивы только для одного типа данных.
3. Почему для конкатенации множества строк в языке Java следует использовать StringBuilder (выберите верные утверждения)?
  - Конкатенация строк оператором «+» не предусмотрена.
  - При конкатенации строк с помощью оператором «+» результат всегда печатается в консоль (стандартный поток вывода – stdout).
  - Конкатенация строк оператором «+» приводит к созданию множества экземпляров строк и многократному копированию данных.
  - Строки не являются ссылочным типом данных.

4. Что возвращает функция, приведенная ниже:

```
public int getXYZ(int[] arr) {
    int a = -1;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] > 0 && (a < 0 || arr[i] < a)) {
            a = arr[i];
        }
    }
    return a;
}
```

- последнее положительное значение в массиве;
- максимальное значение в массиве;
- минимальное значение после первого положительного значения в массиве;
- минимальное положительное значение в массиве.

5. Какие числа встретятся среди всех напечатанных при выполнении следующего фрагмента кода (требуется указать все правильные варианты):

```
int a = 2, b = 0;
for (int i = 0; i < 20; i++) {
    System.out.println(a);
    a += 7;
    if (a >= 20) {
        a -= 20;
        b++;
        if (b >= 3)
            break;
    }
}
```

- 15
- 16
- 17
- 18

6. Строгая типизация предполагает (выберите верные утверждения)?

- Все используемые в функции переменные должны объявляться строго до остального кода функции.
- При компиляции программы весь код (все операции) проверяется на совместимость или возможность преобразования типов, несовместимость считается ошибкой.
- В программе нельзя определить несколько функций с одинаковым именем.
- Язык программирования обязательно должен быть объектно-ориентированным.

7. Определите, что вычисляет следующая функция:

```
public static int solve(int[] arr) {
    int minIndex = 0;
    int maxIndex = 0;
    for (int i = 1; i < arr.length; i++) {
        if (arr[i] < arr[minIndex]) {
            minIndex = i;
        }
        if (arr[i] >= arr[maxIndex]) {
            maxIndex = i;
        }
    }
    return Math.max(Math.abs(minIndex - maxIndex) - 1, 0);
}
```

• количество элементов в массиве между первым минимальным и последним максимальным значением;

- разность между минимальных и максимальным элементом массива;
- количество элементов в массиве между первым максимальным и последним минимальным значением;
- разность между общим количеством минимальных и максимальных элементов в массиве.

8. Выберите верные утверждения для циклов:

- Цикл for выполняется в 2 раза быстрее, чем цикл while.
- Циклы не могут быть использованы в рекурсивных функциях.
- В C-подобных языках (например, Java) любой цикл while формально может быть переписан в виде цикла for.
- Количество вложенных циклов в функции не может быть больше, чем количество параметров в этой функции.

2) открытые задания (тестовые, повышенный уровень сложности):

1. Какая строка будет напечатана в результате выполнения следующего кода?

```
char ch = 'a';
int a = 256;
int b = 0;
while (a > b) {
    System.out.print(ch);
    ch++;
    a = a / 2;
    b++;
}
System.out.println();
```

Ответ: **abcdef**

2. Какое максимальное значение могло храниться в переменной x, если в результате выполнения кода, приведенного ниже, было напечатано число 12?

```
int[] arr = {1, 3, x, 7, 10};
int s = 0;
for (int i = 0; i < arr.length; i++) {
    s += arr[i];
}
s = s / 4;
System.out.println(s);
```

Ответ: **30**

### Критерии и шкалы оценивания заданий ФОС:

#### 1) Задания закрытого типа (выбор одного варианта ответа, верно/неверно):

- 1 балл – указан верный ответ;
- 0 баллов – указан неверный ответ.

#### 2) Задания закрытого типа (множественный выбор):

- 2 балла – указаны все верные ответы;
- 0 баллов — указан хотя бы один неверный ответ.

#### 3) Задания закрытого типа (на соответствие):

- 2 балла – все соответствия определены верно;
- 0 баллов – хотя бы одно сопоставление определено неверно.

#### 4) Задания открытого типа (короткий текст):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

#### 5) Задания открытого типа (число):

- 2 балла – указан верный ответ;

0 баллов – указан неверный ответ.

**Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).**