


МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

**УТВЕРЖДАЮ**

Заведующий кафедрой  
Программирования и информационных технологий

  
\_\_\_\_\_ проф. Махортов С.Д,  
10.03.2021

## **РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.О.17 Программирование

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

09.03.03 Прикладная информатика

**2. Профиль подготовки/специализация:**

Прикладная информатика в экономике

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

**4. Форма обучения:** Очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

Программирования и информационных технологий

**6. Составители программы:**

ст. преподаватель каф. ПиИТ Соломатин Дмитрий Иванович

e-mail: solomatin@cs.vsu.ru

факультет: Компьютерных наук

кафедра: Программирования и информационных технологий

**7. Рекомендована:**

НМС ф-та компьютерных наук, протокол № 5 от 10.03.2021

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**8. Учебный год:** 2021-2022

**Семестр(ы):** 1, 2

## 9. Цели и задачи учебной дисциплины:

Изучение студентами основ программирования и принципов проектирования программ, а также овладение практическими навыками написания относительно простых программ (на конкретном языке).

## 10. Место учебной дисциплины в структуре ООП:

Учебная дисциплина относится к обязательной части блока Б1.

Для успешного освоения дисциплины необходимы знания математики и основ информатики в объеме школьной программы.

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

| Код   | Название компетенции   | Код(ы)  | Индикатор(ы)  | Планируемые результаты обучения   |
|-------|--|---------|---|---|
| ОПК-2 | Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности | ОПК-2.1 | Знает современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности                  | Знать:<br>возможности современных языков программирования (на примере конкретного языка) при решении задач профессиональной деятельности                  |
|       |  | ОПК-2.2 | Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности         | Уметь:<br>выбирать средства языка программирования и использовать возможности современных сред разработки при решении задач профессиональной деятельности |
|       |  | ОПК-2.3 | Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности | Владеть:<br>современным языком программирования на достаточном уровне, чтобы решать задачи профессиональной деятельности приемлемого уровня сложности     |
| ОПК-4 | Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной  | ОПК-4.1 | Знает основные стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы   | Знать:<br>стандарты кодирования для конкретного языка программирования  |

|       |  |         |   |   |
|-------|--|---------|---|---|
|       | деятельностью  | ОПК-4.2 | Умеет применять стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы  | Уметь:<br>правильно оформлять код, писать комментарии в стандарте JavaDoc (или другом общепринятом стандарте) для автоматического построения документации по коду |
|       |  | ОПК-4.3 | Владеет навыками составления технической документации на различных этапах жизненного цикла информационной системы   | Владеть:<br>навыками составления технической документации к разработанному программному продукту  |
| ОПК-7 | Способен разрабатывать алгоритмы и программы, пригодные для практического применения | ОПК-7.1 | Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий  | Знать:<br>основные алгоритмические конструкции языка программирования и принципы алгоритмизации задач   |
|       |  | ОПК-7.2 | Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ | Уметь:<br>применять средства разработки (язык программирования и среду разработки) при реализации практических задач в области информационных систем и технологий |
|       |  | ОПК-7.3 | Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач  | Владеть:<br>навыками программирования для решения практических задач, а также их тестирования и отладки   |

**12. Объем дисциплины в зачетных единицах/час.** (в соответствии с уч. планом) – 8 / 288.

## Форма промежуточной аттестации – Экзамен

### 13. Виды учебной работы

| Вид учебной работы  |              | Трудоемкость |              |        |   |
|---|--------------|--------------|--------------|--------|---|
|   |              | Всего        | По семестрам |        |   |
|   |              |              | 1 сем.       | 2 сем. | – |
| Аудиторные занятия  |              | 96           | 48           | 48     | – |
| в том числе:  | лекции       | 64           | 34           | 34     | – |
|   | практические | 32           | 16           | 16     | – |
|   | лабораторные | –            | –            | –      | – |
| Самостоятельная работа  |              | 120          | 58           | 58     | – |
| в том числе: курсовая работа (проект)                               |              | –            | –            | –      | – |
| Форма промежуточной аттестации (зачет – 0 час. / экзамен – 72 час.) |              | 72           | 36           | 36     | – |
| Итого:  |              | 288          | 144          | 144    | – |

#### 13.1. Содержание дисциплины

| № п/п            | Наименование раздела дисциплины                | Содержание раздела дисциплины   | Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК |
|------------------|--|---|--|
| <b>1. Лекции</b> |  |   |  |
| 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 Java-программ; 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 |  |
| 1.3              | Функции  | Понятие функций (статических методов класса применительно к Java): описание и вызов, передача параметров; проектирование программы с использованием функций; важность разделения программы на подзадачи и правильного выделения подзадач, структуризация и принципы повторного использования кода; примеры программ с использованием функций  |  |
| 1.4              | Операторы управления ходом выполнения программ | 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  |  |
| 1.5              | Простейшие алгоритмы                           | 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                           |  |

|      |  |   |  |
|------|--|---|--|
| 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>).</p> <p>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>   |  |
| 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 <code>Arrays.sort</code>, 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 объектно-ориентированного подхода   | <p>Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между <code>static</code>-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 <code>Object</code> и его методы</p>  |  |
| 1.10 | Создание приложений с оконным интерфейсом  | <p>Принципы создания приложений с оконным интерфейсом; библиотека <code>Swing</code> и базовые <code>Swing</code>-компоненты; визуальное проектирование форм (<code>JFrame</code>) в среде разработки, обработка событий компонентов; понятие модели данных для сложных компонентов, <code>JTable</code> и разработка класса <code>JTableUtils</code> для упрощения работы с <code>JTable</code>.</p> <p>Типичная архитектура приложений с оконным интерфейсом и важность разделения логики и отображения, приложения с несколькими формами, примеры приложений.</p> <p>Второй вариант построения оконного интерфейса - с помощью <code>JavaFX</code>, возможности и особенности, примеры приложений.</p> |  |
| 1.11 | Коллекции                                  | <p>Понятие коллекций, какие виды коллекций бывают (списки, словари, множества, стеки и очереди), иерархия коллекций в Java; списки (<code>List&lt;T&gt;</code>) по сравнению с массивами, методы списков; словари (<code>Map&lt;K, V&gt;</code>) и множества (<code>Set&lt;T&gt;</code>) и их методы; понятие обобщенных типов данных и кода (<code>generics</code>); разница между интерфейсом и реализацией (<code>List&lt;T&gt;</code> и <code>ArrayList&lt;T&gt;/LinkedList&lt;T&gt;</code>, <code>Map&lt;K, V&gt;</code> и <code>TreeMap&lt;K, V&gt;/HashMap&lt;K, V&gt;</code>); примеры эффективного решения задач с помощью коллекций</p>   |  |

|      |  |   |  |
|------|--|---|--|
| 1.12 | 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 применением ООП-подхода - выделение логики в виде отдельного класса, реализация отображения в JTable; знакомство с классом <code>java.awt.Graphics</code>  |  |
| 1.13 | 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 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 ( <code>Stack&lt;T&gt;</code> ) и очередей ( <code>Queue&lt;T&gt;</code> ) |  |
| 1.14 | Основы объектно-ориентированного программирования (инкапсуляция) | Принципы объектно-ориентированного подхода; понятие класса и его экземпляров; различия между <code>static</code> -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 <code>Object</code> и его методы  |  |
| 1.15 | Понятие сложности алгоритма                                      | Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в $O$ -нотации, примеры алгоритмов с различной сложностью   |  |
| 1.16 | Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$       | Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки; простые сортировки и их свойства: пузырьковая сортировка, сортировка выбором, сортировка вставками; реализация указанных сортировок в обобщенном ( <code>generic</code> ) виде с параметром в виде функции сравнения элементов; вызов встроенной функции сортировки   |  |
| 1.17 | «Быстрые» алгоритмы сортировки                                   | Классические «быстрые» ( $O(n \cdot \log(n))$ ) сортировки со сравнением элементов: сортировка слиянием ( <code>merge sort</code> ), быстрая сортировка ( <code>quick sort</code> ), пирамидальная сортировка ( <code>heap sort</code> ); сортировки «без сравнений»: сортировка подсчетом ( <code>counting sort</code> ), блочная сортировка ( <code>bucket sort</code> )  |  |
| 1.18 | Классификация структур данных, обзор коллекций языка Java        | Классификация структур данных по принципу обращения к элементам: записи (классы), массивы, списки, словари, а также стеки и очереди; обзор коллекций Java как реализаций указанных структур данных: <code>List&lt;T&gt;</code> ( <code>ArrayList</code> , <code>LinkedList</code> ), <code>Map&lt;K, V&gt;</code> ( <code>TreeMap</code> , <code>HashMap</code> ), <code>Stack&lt;T&gt;</code> , <code>Queue&lt;T&gt;</code> , практические примеры их применения   |  |
| 1.19 | Односвязные и двусвязные списки, стеки, очереди                  | Архитектура односвязных и двусвязных списков, основные операции; реализация указанных структур данных в виде классов с использованием инкапсуляции; реализация стека и очереди на основе односвязного списка  |  |

|                                |  |   |  |
|--------------------------------|--|---|--|
| 1.20                           | Деревья, двоичные деревья поиска   | Организация данных в виде дерева, терминология, обход и способы отображения деревьев, двоичные деревья; двоичные деревья поиска: идея и основные операции, реализация в виде классов, понятие сбалансированности дерева; реализация множества (Set<T>) и словаря (Map<T>) с помощью двоичного дерева поиска   |  |
| 1.21                           | AVL-деревья  | AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций  |  |
| 1.22                           | 2-3-, 2-3-4- и красно-черные деревья   | 2-3- и 2-3-4- деревья: идея и основные операции; красно-черные деревья как вариант представления 2-3-4-деревьев, основные операции для красно-черных деревьев и оценка сложности  |  |
| 1.23                           | Хеширование (hash-таблицы)   | Hash-таблицы как вариант структуры с потенциальным временем доступа по ключу $O(1)$ , понятие hash-функции и ее качество, стратегии разрешения коллизий – открытая и закрытая адресации, реализация множества (Set<T>) и словаря (Map<T>) с помощью hash-таблиц, сравнение с деревьями поиска; комбинации hash-таблиц с деревьями поиска против «плохих» hash-функций |  |
| 1.24                           | Основы теории графов   | Графы: основные понятия и определения; способы задания (хранения) графов (матрица смежности, списки смежности) и универсальный интерфейс представления графов для алгоритмов  |  |
| 1.25                           | Обход графов, поиск в глубину, поиск в ширину                                      | Реализация обхода графов (поиска) в глубину и ширину; отображение графов с помощью библиотеки Graphviz  |  |
| 1.26                           | Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево) | Поиск кратчайших путей в графе: алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла; поиск минимального остовного дерева: алгоритм Прима, алгоритм Краскала  |  |
| 1.27                           | Примеры практических задач на графах   | Эйлеров граф, Гамильтонов граф; примеры практических задач, которые сводятся к задачам на графах  |  |
| <b>2. Практические занятия</b> |  |   |  |
| 2.1                            | 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                         |  |
| 2.2                            | Функции  | Понятие функций (статических методов класса применительно к Java): описание и вызов, передача параметров; проектирование программы с использованием функций; важность разделения программы на подзадачи и правильного выделения подзадач, структуризация и принципы повторного использования кода; примеры программ с использованием функций                          |  |
| 2.3                            | Операторы управления ходом выполнения программ                                     | 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  |  |

|     |  |  |  |
|-----|--|--|--|
| 2.4 | Простейшие алгоритмы                       | 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  |  |
| 2.5 | Составные типы данных                      | Массивы и множественные массивы (массивы массивов); типичные задачи обработки массивов, класс java.util.Arays; разработка библиотеки функций ввода/вывода массивов и других функции в виде класс ArrayUtils.<br>Перечисления (enum).<br>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.<br>Понятие типов-значений (value types) и ссылочных типов (reference types) данных в Java, понятие объектов, ссылочная модель и сборка мусора. |  |
| 2.6 | Строки                                     | Строки в Java, особенности класса String - особенности реализации (неизменяемость), поддерживаемые методы, сравнение строк с помощью equals.<br>Особенности конкатенации строк и класс StringBuilder.<br>Поверхностное знакомство с регулярными выражения и возможностями их применения (RegExp и некоторые методы String).  |  |
| 2.7 | Типичные задачи обработки массивов и строк | 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  |  |
| 2.8 | Основы объектно-ориентированного подхода   | 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  |  |
| 2.9 | Создание приложений с оконным интерфейс    | 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.<br>Типичная архитектура приложений с оконным интерфейсом и важность разделения логики и отображения, приложения с несколькими формами, примеры приложений.<br>Второй вариант построения оконного интерфейса - с помощью JavaFX, возможности и особенности, примеры приложений.                  |  |



|      |  |   |  |
|------|--|---|--|
| 2.10 | Коллекции  | Понятие коллекций, какие виды коллекций бывают (списки, словари, множества, стеки и очереди), иерархия коллекций в Java; списки ( <code>List&lt;T&gt;</code> ) по сравнению с массивами, методы списков; словари ( <code>Map&lt;K, V&gt;</code> ) и множества ( <code>Set&lt;T&gt;</code> ) и их методы; понятие обобщенных типов данных и кода (generics); разница между интерфейсом и реализацией ( <code>List&lt;T&gt;</code> и <code>ArrayList&lt;T&gt;/LinkedList&lt;T&gt;</code> , <code>Map&lt;K, V&gt;</code> и <code>TreeMap&lt;K, V&gt;/HashMap&lt;K, V&gt;</code> ); примеры эффективного решения задач с помощью коллекций                              |  |
| 2.11 | 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 применением ООП-подхода - выделение логики в виде отдельного класса, реализация отображения в <code>JTable</code> ; знакомство с классом <code>java.awt.Graphics</code>  |  |
| 2.12 | 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 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 ( <code>Stack&lt;T&gt;</code> ) и очередей ( <code>Queue&lt;T&gt;</code> ) |  |
| 2.13 | Основы объектно-ориентированного программирования (инкапсуляция) | Принципы объектно ориентированного подхода; понятие класса и его экземпляров; различия между <code>static</code> -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 <code>Object</code> и его методы  |  |
| 2.14 | Понятие сложности алгоритма                                      | Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в <code>O</code> -нотации, примеры алгоритмов с различной сложностью  |  |
| 2.15 | Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$       | Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки; простые сортировки и их свойства: пузырьковая сортировка, сортировка выбором, сортировка вставками; реализация указанных сортировок в обобщенном (generic) виде с параметром в виде функции сравнения элементов; вызов встроенной функции сортировки  |  |
| 2.16 | «Быстрые» алгоритмы сортировки                                   | Классические «быстрые» ( $O(n \cdot \log(n))$ ) сортировки со сравнением элементов: сортировка слиянием ( <code>merge sort</code> ), быстрая сортировка ( <code>quick sort</code> ), пирамидальная сортировка ( <code>heap sort</code> ); сортировки «без сравнений»: сортировка подсчетом ( <code>counting sort</code> ), блочная сортировка ( <code>bucket sort</code> )  |  |

|                               |  |  |  |
|-------------------------------|--|--|--|
| 2.17                          | Классификация структур данных, обзор коллекций языка Java                          | Классификация структур данных по принципу обращения к элементам: записи (классы), массивы, списки, словари, а также стеки и очереди; обзор коллекций Java как реализаций указанных структур данных: List<T> (ArrayList, LinkedList), Map<K, V> (TreeMap, HashMap), Stack<T>, Queue<T>, практические примеры их применения  |  |
| 2.18                          | Односвязные и двусвязные списки, стеки, очереди                                    | Архитектура односвязных и двусвязных списков, основные операции; реализация указанных структур данных в виде классов с использованием инкапсуляции; реализация стека и очереди на основе односвязного списка   |  |
| 2.19                          | Деревья, двоичные деревья поиска   | Организация данных в виде дерева, терминология, обход и способы отображения деревьев, двоичные деревья; двоичные деревья поиска: идея и основные операции, реализация в виде классов, понятие сбалансированности дерева; реализация множества (Set<T>) и словаря (Map<T>) с помощью двоичного дерева поиска  |  |
| 2.20                          | AVL-деревья  | AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций   |  |
| 2.21                          | 2-3-, 2-3-4- и красно-черные деревья   | 2-3- и 2-3-4- деревья: идея и основные операции; красно-черные деревья как вариант представления 2-3-4-деревьев, основные операции для красно-черных деревьев и оценка сложности   |  |
| 2.22                          | Хеширование (hash-таблицы)   | Hash-таблицы как вариант структуры с потенциальным временем доступа по ключу O(1), понятие hash-функции и ее качество, стратегии разрешения коллизий – открытая и закрытая адресации, реализация множества (Set<T>) и словаря (Map<T>) с помощью hash-таблиц, сравнение с деревьями поиска; комбинации hash-таблиц с деревьями поиска против «плохих» hash-функций |  |
| 2.23                          | Основы теории графов   | Графы: основные понятия и определения; способы задания (хранения) графов (матрица смежности, списки смежности) и универсальный интерфейс представления графов для алгоритмов   |  |
| 2.24                          | Обход графов, поиск в глубину, поиск в ширину                                      | Реализация обхода графов (поиска) в глубину и ширину; отображение графов с помощью библиотеки Graphviz   |  |
| 2.25                          | Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево) | Поиск кратчайших путей в графе: алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла; поиск минимального остовного дерева: алгоритм Прима, алгоритм Краскала   |  |
| 2.26                          | Примеры практических задач на графах   | Эйлеров граф, Гамильтонов граф; примеры практических задач, которые сводятся к задачам на графах   |  |
| <b>3. Лабораторные работы</b> |  |  |  |
| 3.1                           | нет  |  |  |

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

| № п/п | Наименование темы (раздела) дисциплины      | Виды занятий (часов) |              |              |                        |       |
|-------|---|----------------------|--------------|--------------|------------------------|-------|
|       |   | Лекции               | Практические | Лабораторные | Самостоятельная работа | Всего |
| 1     | Введение в предмет                          | 2                    | –            | –            | 3                      | 5     |
| 2     | Переменные и типы данных, ввод-вывод данных | 2                    | 1            | –            | 4                      | 7     |
| 3     | Функции                                     | 3                    | 1            | –            | 4                      | 8     |

|    |  |    |    |   |     |     |
|----|--|----|----|---|-----|-----|
| 4  | Операторы управления ходом выполнения программ                                     | 3  | 2  | – | 6   | 11  |
| 5  | Простейшие алгоритмы   | 2  | 1  | – | 4   | 7   |
| 6  | Составные типы данных  | 3  | 1  | – | 4   | 8   |
| 7  | Строки   | 2  | 1  | – | 4   | 7   |
| 8  | Типичные задачи обработки массивов и строк   | 2  | 2  | – | 6   | 10  |
| 9  | Основы объектно-ориентированного подхода   | 3  | 1  | – | 4   | 8   |
| 10 | Создание приложений с оконным интерфейсом  | 3  | 2  | – | 6   | 11  |
| 11 | Коллекции  | 2  | 1  | – | 4   | 7   |
| 12 | Создание прикладных приложений (создание игры)                                     | 3  | 2  | – | 6   | 11  |
| 13 | Рекурсия и рекурсивные алгоритмы   | 2  | 1  | – | 5   | 8   |
| 14 | Основы объектно-ориентированного программирования (инкапсуляция)                   | 2  | 1  | – | 5   | 8   |
| 15 | Понятие сложности алгоритма  | 2  | 1  | – | 4   | 7   |
| 16 | Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$                         | 2  | 1  | – | 3   | 6   |
| 17 | «Быстрые» алгоритмы сортировки   | 2  | 1  | – | 4   | 7   |
| 18 | Классификация структур данных, обзор коллекций языка Java                          | 2  | 1  | – | 4   | 7   |
| 19 | Односвязные и двусвязные списки, стеки, очереди                                    | 2  | 1  | – | 4   | 7   |
| 20 | Деревья, двоичные деревья поиска   | 2  | 1  | – | 4   | 7   |
| 21 | AVL-деревья  | 4  | 1  | – | 3   | 8   |
| 22 | 2-3-, 2-3-4- и красно-черные деревья   | 4  | 1  | – | 3   | 8   |
| 23 | Хеширование (hash-таблицы)   | 2  | 1  | – | 4   | 7   |
| 24 | Основы теории графов   | 2  | 1  | – | 4   | 7   |
| 25 | Обход графов, поиск в глубину, поиск в ширину                                      | 2  | 1  | – | 4   | 7   |
| 26 | Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево) | 4  | 2  | – | 5   | 11  |
| 27 | Примеры практических задач на графах   | 4  | 2  | – | 5   | 11  |
|    | Итого:   | 68 | 32 | 0 | 116 | 216 |

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

Рекомендуется работа с конспектами лекций, презентационным материалом, выполнение всех лабораторных и контрольных работ, заданий текущей аттестации. Учебные и методические материалы по дисциплине размещены на сетевом диске, доступным на любом компьютере в локальной сети ФКН.

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

а) основная литература:

| № 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.            |
| 4 | Седжвик, Роберт. Фундаментальные алгоритмы на Java : Пер. с англ. / Р. Седжвик .— М. : DiaSoft, 2003. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2003 .— 680 с. : ил. — Библиогр. в конце глав. - Предм. указ.: с. 680-687 .— Парал. тит. л. англ. — ISBN 966-7992-22-5.  |
| 5 | 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 985-5-496-00740-5. |
| 6 | Кормен, Томас. Алгоритмы : Построение и анализ : [Учебник] / Т. Кормен, Ч. Лейзерсон, Р. Ривест ; Пер. с англ. К. Белов и др.; Науч. ред. А. Шень .— М. : МЦНМО, 2002 .— 955 с. : ил. — (Классические учебники: computer science) .— ISBN 5-900916-37-5.   |
| 7 | Ахо, Альфред В. Структуры данных и алгоритмы : [Учебное пособие] / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман; Пер. с англ. и ред. А.А. Минько .— М. и др. : Вильямс, 2003 .— 382 с. : ил., табл. — Парал. тит. л. англ. — Библиогр.: с.369-374 .— Предм. указ.: с. 375-382 .— ISBN 5-8459-0122-7.                  |

б) дополнительная литература:

| № п/п | Источник  |
|-------|---|
| 8     | 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.   |
| 9     | 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.  |
| 10    | Портянкин, Иван Александрович. Swing. Эффектные пользовательские интерфейсы. Java Foundation Classes / Иван Портянкин .— СПб. [и др.] : Питер, 2005 .— 523 с. — (Библиотека программиста) .— Алф. указ.: с.515-523 .— ISBN 5-469-00005-2.   |
| 11    | Дейтел, Х.М. Как программировать на Java / Х.М. Дейтел, П.Д. Дейтел .— М. : Бином, 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.  |
| 12    | Технологии программирования на Java 2 / Х.М. Дейтел, П.Д. Дейтел, С.И. Сантри .— М. : Бином, 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. |
| 13    | Седжвик, Роберт. Фундаментальные алгоритмы на C++ / Р. Седжвик .— М. и др. : DiaSoft, 2002. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2002 .— 687 с. : ил. — ISBN 5-93772-047-4.  |
| 14    | Седжвик, Роберт. Фундаментальные алгоритмы на C++ / Р. Седжвик .— М. и др. : DiaSoft, 2002. — Ч.5: Алгоритмы на графах .— 2002 .— 484 с. : ил. — Парал. тит. л. англ. — ISBN 5-93772-054-7 .— ISBN 0-201-36118-3.   |
| 15    | Седжвик, Роберт. Фундаментальные алгоритмы на C : Пер. с англ. / Р. Седжвик ; Принстонский ун-т .— 3-е изд. — СПб. и др. : DiaSoft, 2003. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2003 .— 670 с. : ил. — Парал. тит. л. англ. — ISBN 5-93772-081-4.                   |
| 16    | Седжвик, Роберт. Фундаментальные алгоритмы на C : Пер. с англ. / Р. Седжвик ; Принстонский ун-т .— 3-е изд. — СПб., и др. : DiaSoft, 2003. — Ч.5: Алгоритмы на графах .— 2003 .— С.661-1127 : ил .— Парал. тит. л. англ. — ISBN 5-93772-082-2.  |

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет):

| № п/п | Источник   |
|-------|--|
| 17    | 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> |
| 18    | Учебник: программирование на Java [Электронный ресурс] : — Режим доступа: <a href="https://java9.ru/">https://java9.ru/</a>  |
| 19    | Иллюстрированный самоучитель по Java [Электронный ресурс] : — Режим доступа: <a href="http://www.realcoding.net/teach/java/">http://www.realcoding.net/teach/java/</a>   |
| 20    | MAXimal::algo [Электронный ресурс] : — Режим доступа: <a href="http://e-maxx.ru/algo/">http://e-maxx.ru/algo/</a>  |

|    |   |
|----|---|
| 21 | Алгоритмы: теория и практика. Методы [Электронный ресурс] : — Режим доступа: <a href="https://stepik.org/course/217/promo">https://stepik.org/course/217/promo</a>                                |
| 22 | Алгоритмы программирования и структуры данных [Электронный ресурс] : — Режим доступа: <a href="https://openedu.ru/course/ITMOUniversity/PADS/">https://openedu.ru/course/ITMOUniversity/PADS/</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. |

## 17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

| № п/п | Наименование  |
|-------|---|
| 1     | OpenJDK - бесплатен   |
| 2     | Среда разработки NetBeans или IntelliJ IDEA (академическая лицензия или версия Community) - бесплатны |

## 18. Материально-техническое обеспечение дисциплины:

| № п/п | Наименование  |
|-------|---|
| 1     | Мультимедийная лекционная аудитория (корп. 1а, ауд. № 479 или другая подходящая): рабочее место преподавателя: ПК-Intel-i3, проектор, видеокоммутатор, микрофон, аудиосистема, специализированная мебель: доски меловые 2 шт., столы и стулья/лавки в количестве, достаточном для размещения потока студентов; выход в Интернет, доступ к фондам учебно-методической документации и электронным изданиям. |
| 2     | Компьютерный класс (корп. 1а, ауд. № 382-385 или другие подходящие): ПК-Intel-i3 16 шт., специализированная мебель: доска маркерная 1 шт., столы и стулья в количестве, достаточном для размещения академической группы (подгруппы) студентов; выход в Интернет, доступ к фондам учебно-методической документации и электронным изданиям.   |

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

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

| № п/п | Наименование раздела дисциплины (модуля)       | Компетенция(и)      | Индикатор(ы) достижения компетенции   | Оценочные средства  |
|-------|--|---------------------|---|---|
| 1     | Введение в предмет                             | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 2     | Переменные и типы данных, ввод-вывод данных    | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 3     | Функции  | ОПК-2, ОПК-4, ОПК-7 | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-7.1, ОПК-7.2, ОПК-7.3 | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 4     | Операторы управления ходом выполнения программ | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 5     | Простейшие алгоритмы                           | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |

|    |  |                     |   |   |
|----|--|---------------------|---|---|
| 6  | Составные типы данных  | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 7  | Строки   | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 8  | Типичные задачи обработки массивов и строк                       | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 9  | Основы объектно-ориентированного подхода                         | ОПК-2, ОПК-4, ОПК-7 | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-7.1, ОПК-7.2, ОПК-7.3 | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 10 | Создание приложений с оконным интерфейсом                        | ОПК-2, ОПК-4, ОПК-7 | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-7.1, ОПК-7.2, ОПК-7.3 | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 11 | Коллекции  | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 12 | Создание прикладных приложений (создание игры)                   | ОПК-2, ОПК-4, ОПК-7 | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-7.1, ОПК-7.2, ОПК-7.3 | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 13 | Рекурсия и рекурсивные алгоритмы                                 | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 14 | Основы объектно-ориентированного программирования (инкапсуляция) | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 15 | Понятие сложности алгоритма                                      | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 16 | Алгоритмы сортировки: свойства, простые алгоритмы $O(n^2)$       | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 17 | «Быстрые» алгоритмы сортировки                                   | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 18 | Классификация структур данных, обзор коллекций языка Java        | ОПК-2, ОПК-4, ОПК-7 | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-7.1, ОПК-7.2, ОПК-7.3 | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 19 | Односвязные и двусвязные списки, стеки, очереди                  | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 20 | Деревья, двоичные деревья поиска                                 | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 21 | AVL-деревья  | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 22 | 2-3-, 2-3-4- и красно-черные деревья                             | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |

|   |  |                     |   |   |
|---|--|---------------------|---|---|
| 23  | Хеширование (hash-таблицы)   | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 24  | Основы теории графов   | ОПК-2, ОПК-4, ОПК-7 | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-7.1, ОПК-7.2, ОПК-7.3 | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 25  | Обход графов, поиск в глубину, поиск в ширину                                      | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 26  | Оптимизационные алгоритмы на графах (кратчайшие пути, минимальное остовное дерево) | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| 27  | Примеры практических задач на графах   | ОПК-2, ОПК-7        | ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3                            | Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения) |
| Промежуточная аттестация<br>форма контроля – экзамены (в 1-ом и во 2-ом семестре) |  |                     |   | Перечень вопросов к экзамену из пункта 20.2                                     |

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

### 20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью контроля выполнения обязательных практических заданий.

Перечень практических заданий в 1-ом семестре:

| № п/п | Задание  |
|-------|--|
| 1     | Задача 1 - Запись выражений и оператор присваивания ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН) |
| 2     | Задача 2 - Условный оператор ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                        |
| 3     | Задача 3 - Применение функций ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                       |
| 4     | Задача 4 - Циклы ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                    |
| 5     | Задача 5 - Циклы (рисование фигуры псевдографикой, $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)    |
| 6     | Задача 6 - Циклы (вычисление суммы ряда, $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)              |
| 7     | Задача 7 - Одномерные массивы ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                       |
| 8     | Задача 8 - Двумерные массивы ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                        |
| 9     | Задача 9 - Коллекции ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                |
| 10    | Задача 10 - Структуры данных ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                        |
| 11    | Задача 11 - Строки ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                  |
| 12    | Задача 12 - Рекурсия ( $\geq 20$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                |
| 13    | Задача 13 - Логическая игра ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                         |

Перечень практических заданий во 2-ом семестре:

| № п/п | Задание   |
|-------|---|
| 1     | Задача 1 – Инкапсуляция ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                |
| 2     | Задача 2 – Связные списки ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                              |
| 3     | Задача 3 – Стеки и очереди ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                             |
| 4     | Задача 4 – Сортировки ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                  |
| 5     | Задача 5 – Деревья ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                     |
| 6     | Задача 6 – Словари (на основе деревьев или hash-функций, $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН) |
| 7     | Задача 7 – Графы 1 ( $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)                                     |
| 8     | Задача 8 – Графы 2 (интерактивное редактирование графа, $\geq 30$ индивидуальных вариантов, размещены на общедоступном диске в сети ФКН)  |

## 20.2 Промежуточная аттестация

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

1) знание теоретических основ учебного материала, основных определений, понятий и используемой терминологии;

2) умение проводить обоснование и представление основных теоретических и практических результатов (теорем, алгоритмов, методик) с использованием математических выкладок, блок-схем, 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».

Соотношение показателей, критериев и шкалы оценивания результатов обучения на экзамене представлено в следующей таблице.

| Критерии оценивания компетенций   | Уровень сформированности компетенций | Шкала оценок |
|---|--------------------------------------|--------------|
| Студент владеет основными понятиями учебной дисциплины, может пояснить большинство принципов на примерах; вовремя сдал все практические задания, которые выполнены на высоком уровне, без явных ошибок.                 | Повышенный уровень                   | Отлично      |
| Студент владеет основными понятиями учебной дисциплины, однако в ответах на некоторые вопросы допускает неточности; сдал все практические задания, однако к некоторым решениям студента у преподавателя есть замечания. | Базовый уровень                      | Хорошо       |



|   |                   |                     |
|---|-------------------|---------------------|
| Студент знает основные определения из учебной дисциплины, однако пояснить многие понятия на примерах затрудняется; сдал большую часть практических заданий, однако продемонстрированные решения содержат существенные ошибки. | Пороговый уровень | Удовлетворительно   |
| Студент путается в основных понятиях учебной дисциплины, не может привести примеры; не сдал большую часть практических заданий.   | –                 | Неудовлетворительно |

Перечень вопросов к экзамену в 1-ом семестре:

| № п/п | Вопрос  |
|-------|---|
| 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                  |

Перечень вопросов к экзамену во 2-ом семестре:

| № п/п | Вопрос   |
|-------|--|
| 1     | Понятие сложности алгоритма  |
| 2     | Практические приемы оценки сложности алгоритмов  |
| 3     | Сортировка слиянием (merge sort): идея, реализация, оценка сложности   |
| 4     | Быстрая сортировка (quick sort): идея, реализация, оценка сложности  |
| 5     | Пирамидальная сортировка (heap sort): идея, реализация, оценка сложности   |
| 6     | Сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort)               |
| 7     | Классификация структур данных, обзор классов-коллекций стандартной библиотеки языка Java                         |
| 8     | Принципы реализации структур данных в виде классов с применением инкапсуляции                                    |
| 9     | Односвязные списки   |
| 10    | Двусвязные списки  |
| 11    | Реализация стека и очереди в виде связного списка  |
| 12    | Очередь с приоритетом: идея, возможные варианты эффективной реализации   |
| 13    | Деревья: терминология, обход и способы отображения деревьев, двоичные деревья                                    |
| 14    | Двоичные деревья поиска: основные операции, понятие сбалансированности дерева                                    |
| 15    | Реализация множества (Set) и словаря (Map) на основе двоичного дерева поиска                                     |
| 16    | AVL-деревья: идея, операции вставки и удаления, балансировка (R-/L-, RL-/LR-повороты), оценка сложности операций |
| 17    | 2-3- и 2-3-4- деревья: идея и основные операции, оценка сложности  |
| 18    | Красно-черные деревья: идея и основные операции, оценка сложности  |

|    |  |
|----|--|
| 19 | Hash-таблицы: понятие hash-функции, коллизии и стратегии их разрешения   |
| 20 | Реализация множества (Set) и словаря (Map) на основе hash-таблицы        |
| 21 | Графы: основные понятия и определения, способы задания (хранения) графов |
| 22 | Реализация обхода графов (поиска) в глубину и ширину                     |
| 23 | Отображение графов с помощью библиотеки Graphviz                         |
| 24 | Алгоритм Дейкстры  |
| 25 | Алгоритм Беллмана-Форда  |
| 26 | Алгоритм Флойда-Уоршелла   |
| 27 | Алгоритм Прима   |
| 28 | Алгоритм Краскала  |
| 29 | Примеры практических задач, которые сводятся к задачам на графах         |