

Документ подписан простой электронной подписью  
 Информация о владельце:  
 ФИО: Стрельникова Светлана Евгеньевна  
 Должность: ректор  
 Дата подписания: 28.09.2026 16:46:35  
 Уникальный программный ключ:  
 ed7e7ef871ef28d870011250a92f17921078cc18

**Автономная некоммерческая образовательная организация высшего образования  
 «Контур Университет»**



УТВЕРЖДАЮ

С.Е. Стрельникова

, 2026 г.

**МОДУЛЬ 3. ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
 И ПРОГРАММИРОВАНИЯ  
 Программирование**

**рабочая программа дисциплины (модуля)**

Учебный план 09.03.03\_ИИС\_2026.plx  
 09.03.03 Прикладная информатика

Квалификация **бакалавр**  
 Форма обучения **очная**  
 Общая трудоемкость **5 ЗЕТ**

|                         |     |                            |
|-------------------------|-----|----------------------------|
| Часов по учебному плану | 180 | Виды контроля в семестрах: |
| в том числе:            |     | экзамены 2                 |
| аудиторные занятия      | 82  |                            |
| самостоятельная работа  | 89  |                            |
| часов на контроль       | 9   |                            |

**Распределение часов дисциплины по семестрам**

| Семестр<br>( <b>&lt;Курс&gt;.&lt;Семестр на<br/>курсе&gt;</b> ) | <b>2 (1.2)</b> |     | Итого |     |
|-----------------------------------------------------------------|----------------|-----|-------|-----|
| Неделя                                                          | 17 1/6         |     |       |     |
| Вид занятий                                                     | уп             | рп  | уп    | рп  |
| Лекции                                                          | 22             | 22  | 22    | 22  |
| Практические                                                    | 60             | 60  | 60    | 60  |
| Итого ауд.                                                      | 82             | 82  | 82    | 82  |
| Контактная работа                                               | 82             | 82  | 82    | 82  |
| Сам. работа                                                     | 89             | 89  | 89    | 89  |
| Часы на контроль                                                | 9              | 9   | 9     | 9   |
| Итого                                                           | 180            | 180 | 180   | 180 |

Рабочая программа дисциплины

**Программирование**

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки  
09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

Одобрена Ученым советом вуза

Протокол от 15.09.2026 № 05.

Одобрена учебно-методической комиссией

Протокол от 10.09.2026 г. № 01

| <b>1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.1                                                  | Цель освоения дисциплины:                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 1.2                                                  | Формирование у обучающихся системных знаний об алгоритмах обработки данных, структурах данных и современных языках и парадигмах программирования, а также практических навыков построения, реализации, тестирования, отладки и оптимизации программ, пригодных для практического применения, с обоснованным выбором алгоритмических решений и с использованием современных инструментальных средств и систем контроля версий, в том числе отечественного производства. |
| 1.3                                                  | Задачи изучения дисциплины:                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 1.4                                                  | Сформировать понятийный аппарат алгоритмизации: алгоритм, его свойства, способы описания (словесное описание, блок-схемы, псевдокод, UML-диаграммы деятельности).                                                                                                                                                                                                                                                                                                      |
| 1.5                                                  | Сформировать умение оценивать вычислительную и ёмкостную сложность алгоритмов с использованием асимптотических обозначений.                                                                                                                                                                                                                                                                                                                                            |
| 1.6                                                  | Ознакомить с основными структурами данных (массивы, связанные списки, стеки, очереди, деревья, кучи, хэш-таблицы, графы) и особенностями их применения.                                                                                                                                                                                                                                                                                                                |
| 1.7                                                  | Сформировать навыки реализации базовых алгоритмов обработки данных: сортировки, поиска, обхода графов, рекурсии, динамического программирования, — и умение обоснованно выбирать алгоритм и структуру данных под прикладную задачу.                                                                                                                                                                                                                                    |
| 1.8                                                  | Сформировать знания о современных парадигмах программирования (процедурной, объектно-ориентированной, функциональной) и принципах структурного проектирования программ.                                                                                                                                                                                                                                                                                                |
| 1.9                                                  | Сформировать навыки разработки программ на языке высокого уровня с использованием стандартных библиотек и фреймворков.                                                                                                                                                                                                                                                                                                                                                 |
| 1.10                                                 | Сформировать умение применять объектно-ориентированный подход: проектировать классы, применять наследование, полиморфизм и инкапсуляцию.                                                                                                                                                                                                                                                                                                                               |
| 1.11                                                 | Сформировать навыки разработки пользовательских интерфейсов для консольных, настольных и веб-платформ с учётом принципов UI/UX.                                                                                                                                                                                                                                                                                                                                        |
| 1.12                                                 | Сформировать навыки тестирования, отладки, профилирования и рефакторинга программного кода.                                                                                                                                                                                                                                                                                                                                                                            |
| 1.13                                                 | Ознакомить с программными средствами отечественного производства и особенностями их конфигурирования для решения профессиональных задач.                                                                                                                                                                                                                                                                                                                               |

| <b>2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ</b>                                                 |                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Цикл (раздел) ОП:                                                                                                         | Б1.О.03                                                                                                          |
| <b>2.1 Требования к предварительной подготовке обучающегося:</b>                                                          |                                                                                                                  |
| 2.1.1                                                                                                                     | Высшая математика                                                                                                |
| 2.1.2                                                                                                                     | Введение в профессиональную деятельность                                                                         |
| <b>2.2 Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b> |                                                                                                                  |
| 2.2.1                                                                                                                     | Информационные технологии и сервисы                                                                              |
| 2.2.2                                                                                                                     | Базы данных и SQL                                                                                                |
| 2.2.3                                                                                                                     | Архитектура ЭВМ                                                                                                  |
| 2.2.4                                                                                                                     | Операционные системы                                                                                             |
| 2.2.5                                                                                                                     | Программная инженерия                                                                                            |
| 2.2.6                                                                                                                     | Web-технологии                                                                                                   |
| 2.2.7                                                                                                                     | Интеллектуальный анализ данных. Машинное обучение                                                                |
| 2.2.8                                                                                                                     | Разработка и интеграция информационных систем                                                                    |
| 2.2.9                                                                                                                     | Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)) |
| 2.2.10                                                                                                                    | Производственная практика (технологическая (проектно-технологическая))                                           |

| <b>3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ИНДИКАТОРЫ ИХ ДОСТИЖЕНИЯ</b>                                                             |
|--------------------------------------------------------------------------------------------------------------------------|
| <b>ОПК-5: Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем</b> |
| <b>ОПК-5.2: Умеет выполнять параметрическую настройку информационных и автоматизированных систем</b>                     |

|                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения</b>                                                                                                                                                                                        |
| <b>ОПК-7.1: Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий</b>                                                                                              |
| <b>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</b> |

| <b>4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)</b> |                                                                                                                                                                               |                       |              |                   |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|--------------|-------------------|
| <b>Код занятия</b>                                   | <b>Наименование разделов и тем /вид занятия/</b>                                                                                                                              | <b>Семестр / Курс</b> | <b>Часов</b> | <b>Примечание</b> |
|                                                      | <b>Раздел 1.</b>                                                                                                                                                              |                       |              |                   |
| 1.1                                                  | Языки программирования высокого уровня: классификация, парадигмы, трансляция. Типы данных, переменные, операторы, управляющие конструкции. Среда разработки и отладчик /Лек/  | 2                     | 2            |                   |
| 1.2                                                  | Языки программирования высокого уровня: классификация, парадигмы, трансляция. Типы данных, переменные, операторы, управляющие конструкции. Среда разработки и отладчик /Пр/   | 2                     | 6            |                   |
| 1.3                                                  | Языки программирования высокого уровня: классификация, парадигмы, трансляция. Типы данных, переменные, операторы, управляющие конструкции. Среда разработки и отладчик /Ср/   | 2                     | 8            |                   |
| 1.4                                                  | Понятие алгоритма и его свойства. Способы описания алгоритмов: блок-схемы (ГОСТ 19.701-90), псевдокод, UML-диаграммы деятельности. Базовые алгоритмические конструкции /Лек/  | 2                     | 1            |                   |
| 1.5                                                  | Понятие алгоритма и его свойства. Способы описания алгоритмов: блок-схемы (ГОСТ 19.701-90), псевдокод, UML-диаграммы деятельности. Базовые алгоритмические конструкции /Пр/   | 2                     | 2            |                   |
| 1.6                                                  | Понятие алгоритма и его свойства. Способы описания алгоритмов: блок-схемы (ГОСТ 19.701-90), псевдокод, UML-диаграммы деятельности. Базовые алгоритмические конструкции /Ср/   | 2                     | 4            |                   |
| 1.7                                                  | Процедурное программирование. Функции, области видимости, передача параметров, рекурсия. Модульная структура программы. Принципы структурного проектирования /Лек/            | 2                     | 2            |                   |
| 1.8                                                  | Процедурное программирование. Функции, области видимости, передача параметров, рекурсия. Модульная структура программы. Принципы структурного проектирования /Пр/             | 2                     | 6            |                   |
| 1.9                                                  | Процедурное программирование. Функции, области видимости, передача параметров, рекурсия. Модульная структура программы. Принципы структурного проектирования /Ср/             | 2                     | 8            |                   |
| 1.10                                                 | Анализ алгоритмов. Временная и ёмкостная сложность. Асимптотические обозначения $O$ , $\Omega$ , $\Theta$ . Худший, средний и лучший случай. Амортизированная сложность /Лек/ | 2                     | 1            |                   |

|      |                                                                                                                                                                              |   |   |  |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---|--|
| 1.11 | Анализ алгоритмов. Временная и ёмкостная сложность. Асимптотические обозначения $O$ , $\Omega$ , $\Theta$ . Худший, средний и лучший случай. Амортизированная сложность /Пр/ | 2 | 2 |  |
| 1.12 | Анализ алгоритмов. Временная и ёмкостная сложность. Асимптотические обозначения $O$ , $\Omega$ , $\Theta$ . Худший, средний и лучший случай. Амортизированная сложность /Ср/ | 2 | 4 |  |
| 1.13 | Коллекции и работа с данными: списки, словари, множества, кортежи. Файловый ввод-вывод, форматы CSV и JSON. Обработка исключений /Лек/                                       | 2 | 2 |  |
| 1.14 | Коллекции и работа с данными: списки, словари, множества, кортежи. Файловый ввод-вывод, форматы CSV и JSON. Обработка исключений /Пр/                                        | 2 | 6 |  |
| 1.15 | Коллекции и работа с данными: списки, словари, множества, кортежи. Файловый ввод-вывод, форматы CSV и JSON. Обработка исключений /Ср/                                        | 2 | 8 |  |
| 1.16 | Линейные структуры данных: массивы, динамические массивы, односвязные и двусвязные списки, стеки, очереди, деки. Стоимость операций /Лек/                                    | 2 | 1 |  |
| 1.17 | Линейные структуры данных: массивы, динамические массивы, односвязные и двусвязные списки, стеки, очереди, деки. Стоимость операций /Пр/                                     | 2 | 2 |  |
| 1.18 | Линейные структуры данных: массивы, динамические массивы, односвязные и двусвязные списки, стеки, очереди, деки. Стоимость операций /Ср/                                     | 2 | 4 |  |
| 1.19 | Алгоритмы сортировки (пузырьком, вставками, слиянием, быстрая, поразрядная) и поиска (линейный, бинарный). Устойчивость сортировки /Лек/                                     | 2 | 1 |  |
| 1.20 | Алгоритмы сортировки (пузырьком, вставками, слиянием, быстрая, поразрядная) и поиска (линейный, бинарный). Устойчивость сортировки /Пр/                                      | 2 | 2 |  |
| 1.21 | Алгоритмы сортировки (пузырьком, вставками, слиянием, быстрая, поразрядная) и поиска (линейный, бинарный). Устойчивость сортировки /Ср/                                      | 2 | 4 |  |
| 1.22 | Иерархические и ассоциативные структуры: деревья, двоичные деревья поиска, сбалансированные деревья, кучи, хэш- таблицы, разрешение коллизий /Лек/                           | 2 | 1 |  |
| 1.23 | Иерархические и ассоциативные структуры: деревья, двоичные деревья поиска, сбалансированные деревья, кучи, хэш- таблицы, разрешение коллизий /Пр/                            | 2 | 2 |  |
| 1.24 | Иерархические и ассоциативные структуры: деревья, двоичные деревья поиска, сбалансированные деревья, кучи, хэш- таблицы, разрешение коллизий /Ср/                            | 2 | 4 |  |
| 1.25 | Графы: представление, обходы в глубину и ширину, кратчайшие пути. Рекурсия и динамическое программирование. Жадные алгоритмы /Лек/                                           | 2 | 1 |  |
| 1.26 | Графы: представление, обходы в глубину и ширину, кратчайшие пути. Рекурсия и динамическое программирование. Жадные алгоритмы /Пр/                                            | 2 | 2 |  |
| 1.27 | Графы: представление, обходы в глубину и ширину, кратчайшие пути. Рекурсия и динамическое программирование. Жадные алгоритмы /Ср/                                            | 2 | 4 |  |
| 1.28 | Объектно-ориентированное программирование: классы и объекты, инкапсуляция, наследование, полиморфизм, абстрактные типы. Принципы SOLID /Лек/                                 | 2 | 2 |  |
| 1.29 | Объектно-ориентированное программирование: классы и объекты, инкапсуляция, наследование, полиморфизм, абстрактные типы. Принципы SOLID /Пр/                                  | 2 | 6 |  |
| 1.30 | Объектно-ориентированное программирование: классы и объекты, инкапсуляция, наследование, полиморфизм, абстрактные типы. Принципы SOLID /Ср/                                  | 2 | 8 |  |
| 1.31 | Элементы функционального программирования: функции высшего порядка, замыкания, неизменяемые данные, ленивые вычисления /Лек/                                                 | 2 | 2 |  |

|      |                                                                                                                                                                                                       |   |   |  |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---|--|
| 1.32 | Элементы функционального программирования: функции высшего порядка, замыкания, неизменяемые данные, ленивые вычисления /Пр/                                                                           | 2 | 6 |  |
| 1.33 | Элементы функционального программирования: функции высшего порядка, замыкания, неизменяемые данные, ленивые вычисления /Ср/                                                                           | 2 | 8 |  |
| 1.34 | Тестирование и отладка программ. Уровни тестирования, техники тест-дизайна, модульные тесты, покрытие кода, логирование /Лек/                                                                         | 2 | 2 |  |
| 1.35 | Тестирование и отладка программ. Уровни тестирования, техники тест-дизайна, модульные тесты, покрытие кода, логирование /Пр/                                                                          | 2 | 6 |  |
| 1.36 | Тестирование и отладка программ. Уровни тестирования, техники тест-дизайна, модульные тесты, покрытие кода, логирование /Ср/                                                                          | 2 | 8 |  |
| 1.37 | Качество кода: стиль оформления, документирование, рефакторинг, профилирование и оптимизация. Системы контроля версий и командная работа с кодом /Лек/                                                | 2 | 2 |  |
| 1.38 | Качество кода: стиль оформления, документирование, рефакторинг, профилирование и оптимизация. Системы контроля версий и командная работа с кодом /Пр/                                                 | 2 | 6 |  |
| 1.39 | Качество кода: стиль оформления, документирование, рефакторинг, профилирование и оптимизация. Системы контроля версий и командная работа с кодом /Ср/                                                 | 2 | 8 |  |
| 1.40 | Пользовательские интерфейсы: консольный, графический настольный, веб-интерфейс. Принципы UI/UX, валидация ввода. Окружение проекта, управление зависимостями, отечественные средства разработки /Лек/ | 2 | 2 |  |
| 1.41 | Пользовательские интерфейсы: консольный, графический настольный, веб-интерфейс. Принципы UI/UX, валидация ввода. Окружение проекта, управление зависимостями, отечественные средства разработки /Пр/  | 2 | 6 |  |
| 1.42 | Пользовательские интерфейсы: консольный, графический настольный, веб-интерфейс. Принципы UI/UX, валидация ввода. Окружение проекта, управление зависимостями, отечественные средства разработки /Ср/  | 2 | 9 |  |
| 1.43 | /Экзамен/                                                                                                                                                                                             | 2 | 9 |  |

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 5.1. Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации

Темы для обсуждения на практических занятиях:

Семинар № 1. Парадигмы программирования: выбор под задачу (к теме 1.3)

Цель занятия: сформировать понимание сильных и слабых сторон процедурного, объектно-ориентированного и функционального подходов и умение обоснованно выбирать парадигму.

Вопросы для обсуждения: 1. Понятие парадигмы программирования; мультипарадигменные языки. 2. Процедурный подход: декомпозиция по действиям, границы применимости. 3. Объектно-ориентированный подход: декомпозиция по сущностям, стоимость абстракции. 4. Функциональный подход: чистые функции, неизменяемость, предсказуемость поведения. 5. Критерии выбора парадигмы: размер задачи, изменчивость требований, требования к параллелизму и тестируемости. Практические задания: реализация одной и той же задачи (обработка журнала событий) тремя способами; сравнение решений по объёму кода, читаемости и тестируемости; заполнение сравнительной таблицы; мини-дискуссия о том, какой подход целесообразен для типового ИТ-проекта.

Семинар № 2. Сравнительный анализ линейных структур данных (к теме 1.6)

Цель занятия: сформировать у обучающихся умение сопоставлять структуры данных по стоимости базовых операций и обоснованно выбирать структуру под профиль нагрузки.

Вопросы для обсуждения: 1. Массив и связный список: устройство в памяти, локальность доступа. 2. Стоимость операций вставки, удаления, доступа по индексу и поиска для массива, списка, стека, очереди, дека. 3. Динамический массив: амортизированная сложность добавления элемента. 4. Когда список выигрывает у массива и почему на практике это происходит реже, чем следует из теории. 5. Выбор структуры данных по профилю операций.

Практические задания: заполнение сравнительной таблицы «структура данных × операция → сложность»; разбор трёх прикладных ситуаций (буфер событий, история операций с отменой, очередь заданий) с обоснованием выбора структуры; защита выбора в мини-дискуссии «массив против списка».

Семинар № 3. Алгоритмы на графах и динамическое программирование (к теме 1.9)

Цель занятия: сформировать умение сводить прикладную задачу к задаче на графе и выбирать корректный алгоритм её решения.

Вопросы для обсуждения: 1. Способы представления графа: матрица смежности, список смежности, матрица инцидентности. 2. Обходы в глубину и ширину: свойства, области применения, сложность. 3. Задача о кратчайшем пути: алгоритмы Дейкстры и Флойда — Уоршелла, ограничения по знаку весов. 4. Рекурсия и её стоимость; переход от рекурсии к мемоизации и табличному динамическому программированию. 5. Признаки задач, допускающих решение методом динамического программирования (оптимальная подструктура, перекрывающиеся подзадачи).

Практические задания: моделирование сети маршрутов между узлами инфраструктуры и поиск кратчайшего пути; решение задачи о рюкзаке двумя способами (полный перебор и динамическое программирование) с замером времени; обсуждение результатов замеров.

Семинар № 4. Наследование или композиция (к теме 1.10)

Цель занятия: сформировать умение проектировать иерархии типов и осознанно выбирать между наследованием и композицией.

Вопросы для обсуждения: 1. Инкапсуляция и границы ответственности класса. 2. Наследование: отношение «является», проблемы хрупкого базового класса. 3. Композиция и делегирование: отношение «содержит». 4. Полиморфизм и принцип подстановки Лисков. 5. Принципы SOLID и их практическая интерпретация.

Практические задания: разбор трёх предложенных иерархий классов с выявлением нарушений принципа подстановки; переработка одной из иерархий с заменой наследования композицией; защита проектного решения.

Семинар № 5. Инструментальная среда разработчика, в том числе отечественная (к теме 1.14)

Цель занятия: сформировать умение подбирать и настраивать инструментальное окружение разработки с учётом требований к использованию отечественного ПО.

Вопросы для обсуждения: 1. Состав окружения разработки: интерпретатор/компилятор, менеджер пакетов, среда разработки, система контроля версий, средства сборки. 2. Единый реестр российского программного обеспечения: назначение и порядок использования. 3. Отечественные среды исполнения и разработки: возможности и ограничения. 4. Управление зависимостями и воспроизводимость сборки. 5. Риски внешних зависимостей и способы их снижения.

Практические задания: сопоставление двух средств разработки по функциональным критериям; развёртывание рабочего окружения на отечественной операционной системе; фиксация конфигурации проекта в файле зависимостей и проверка воспроизводимости сборки.

Практико-ориентированные задания:

Задание № 1 «Первая программа и работа с отладчиком» (к теме 1.1)

Цель: сформировать навыки разработки и пошаговой отладки программы, ведения истории изменений.

Условие: необходимо разработать программу расчёта итоговой стоимости заказа с учётом скидок по нескольким правилам, применяемым в заданном порядке.

Задания для обучающихся: 1. Реализуйте программу с вводом данных и выводом результата. 2. Инициализируйте репозиторий Git, выполните не менее пяти осмысленных коммитов, создайте ветку для доработки и выполните слияние. 3. Установите точки останова и продемонстрируйте пошаговую отладку с наблюдением значений переменных. 4. Обработайте некорректный ввод пользователя. 5. Оформите README с описанием запуска.

Форма отчётности: ссылка на репозиторий, протокол отладки, устная демонстрация.

Задание № 2 «Формализация алгоритма» (к теме 1.2)

Цель: сформировать навык перевода словесной постановки задачи в формальное описание алгоритма.

Условие: сервис онлайн-записи обрабатывает заявки клиентов. Заявка принимается, если выбранный слот свободен, клиент не имеет более двух активных записей и до начала слота остаётся не менее одного часа. При отказе клиенту предлагается ближайший свободный слот того же дня, а при его отсутствии — первый свободный слот следующего рабочего дня.

Задания для обучающихся: 1. Выделите входные и выходные данные алгоритма, определите предусловия и постусловия. 2. Постройте блок-схему алгоритма в соответствии с ГОСТ 19.701-90. 3. Запишите тот же алгоритм на псевдокоде. 4. Постройте UML-диаграмму деятельности для описанного процесса. 5. Укажите, какие ветви алгоритма требуют обязательного тестирования, и приведите по одному тестовому набору данных на каждую ветвь.

Форма отчётности: практическая работа с приложением блок-схемы, псевдокода и диаграммы деятельности.

Задание № 3 «Модульная программа» (к темам 1.3, 1.5)

Цель: сформировать навыки декомпозиции программы на модули и работы с коллекциями и файлами.

Условие: дан файл журнала обращений пользователей в формате CSV (дата, идентификатор пользователя, категория обращения, длительность, статус) объёмом до 100 000 строк.

Задания для обучающихся: 1. Спроектируйте модульную структуру программы, разделив ввод-вывод, обработку и представление результата. 2. Реализуйте загрузку данных с проверкой корректности строк и обработкой исключений. 3. Реализуйте расчёт статистики: количество обращений по категориям, средняя длительность, доля закрытых обращений, топ -10 пользователей. 4. Реализуйте выгрузку результата в JSON. 5. Обоснуйте выбор коллекций для каждой части расчёта оценкой сложности операций.

Форма отчётности: программный код, файл результата, пояснительная записка.

Задание № 4 «Оценка сложности» (к теме 1.4)

Цель: сформировать навык асимптотической оценки алгоритмов и сопоставления их эффективности.

Условие: предложены пять фрагментов кода: вложенный цикл по двум коллекциям; последовательный поиск в отсортированном массиве; бинарный поиск; рекурсивное вычисление чисел Фибоначчи; то же вычисление с мемоизацией.

Задания для обучающихся: 1. Для каждого фрагмента определите временную сложность в худшем и среднем случае, ёмкостную сложность. 2. Обоснуйте оценку, указав доминирующую операцию. 3. Расположите фрагменты по возрастанию асимптотической сложности. 4. Оцените, при каком объёме входных данных различие между  $O(n \log n)$  и  $O(n^2)$  становится практически значимым; подтвердите вывод расчётом. 5. Предложите способ снижения сложности для двух наименее эффективных фрагментов.

Форма отчётности: письменное заключение с расчётами и таблицей сравнения.

**Задание № 5 «Реализация и тестирование алгоритмов» (к теме 1.7)**

Цель: сформировать навыки программной реализации и тестирования алгоритмов сортировки и поиска.

Условие: дан набор данных объемом до  $10^6$  записей о транзакциях (идентификатор, дата, сумма, статус).

Задания для обучающихся: 1. Реализуйте сортировку вставками, сортировку слиянием и быструю сортировку. 2. Реализуйте линейный и бинарный поиск по ключу. 3. Подготовьте наборы тестовых данных: случайный, отсортированный, обратно отсортированный, с большим числом повторов, пустой и одноэлементный. 4. Измерьте время работы на каждом наборе, оформите результаты в виде таблицы и графика. 5. Объясните расхождения между теоретической оценкой и измеренным временем. 6. Определите, какие из реализованных сортировок являются устойчивыми, и подтвердите это тестом.

Форма отчётности: программный код, протокол тестирования, письменные выводы.

**Задание № 6 «Модульные тесты» (к теме 1.12)**

Цель: сформировать навыки тест-дизайна и автоматизированного тестирования.

Условие: необходимо покрыть тестами модуль расчёта, разработанный в задании № 3.

Задания для обучающихся: 1. Выделите классы эквивалентности и граничные значения для входных данных. 2. Составьте таблицу тест-кейсов с ожидаемым результатом. 3. Реализуйте не менее 15 модульных тестов, включая негативные сценарии. 4. Получите отчёт о покрытии кода и объясните, какие ветви остались непокрытыми и почему. 5. Внесите в код преднамеренную ошибку и покажите, какой тест её обнаруживает.

Форма отчётности: код тестов, таблица тест-кейсов, отчёт о покрытии.

**Задание № 7 «Рефакторинг и профилирование» (к теме 1.13)**

Цель: сформировать навыки повышения качества и производительности кода.

Условие: предоставлен фрагмент кода объёмом около 200 строк, содержащий дублирование, длинные функции, «магические числа» и неэффективную обработку коллекций.

Задания для обучающихся: 1. Выявите и перечислите признаки низкого качества кода. 2. Проведите рефакторинг, применив не менее пяти различных приёмов, и обоснуйте каждый. 3. Подтвердите сохранение поведения программы существующими тестами. 4. Выполните профилирование до и после оптимизации, приведите замеры. 5. Оцените изменение асимптотической сложности критического участка.

Форма отчётности: сравнение версий кода, протокол профилирования, письменные выводы.

**Задание № 8 «Окружение проекта» (к теме 1.14)**

Цель: сформировать навыки конфигурирования воспроизводимого окружения разработки.

Условие: проект должен запускаться на отечественной операционной системе на компьютерах трёх участников команды с одинаковым результатом.

Задания для обучающихся: 1. Создайте изолированное окружение и зафиксируйте состав зависимостей с указанием версий. 2. Опишите порядок развёртывания в документации. 3. Настройте переменные окружения и конфигурационный файл, вынеся из кода все параметры среды. 4. Продемонстрируйте развёртывание «с нуля» на чистой системе. 5. Опишите риски, связанные с обновлением зависимостей, и способы их снижения.

Форма отчётности: конфигурационные файлы, инструкция по развёртыванию, демонстрация.

**Задание № 9 «Пользовательский интерфейс» (к теме 1.14)**

Цель: сформировать навыки проектирования и реализации интерфейса пользователя.

Условие: для программы из задания № 3 требуется интерфейс, доступный неподготовленному пользователю.

Задания для обучающихся: 1. Опишите основные сценарии использования. 2. Спроектируйте интерфейс (эскиз экранов или структура команд). 3. Реализуйте интерфейс в консольном либо графическом варианте. 4. Обеспечьте валидацию ввода и понятные сообщения об ошибках. 5. Оцените решение по критериям наглядности, числа шагов до результата и устойчивости к ошибочным действиям.

Форма отчётности: программный код, эскизы, устная демонстрация.

Примеры кейсовых заданий:

**Кейс № 1 «Выбор структуры данных для сервиса» (к теме 1.8)**

Цель: сформировать навык обоснованного выбора структуры данных под профиль нагрузки.

Фабула: разрабатывается сервис учёта посетителей делового центра. Требуется: мгновенно проверять наличие пропуска по номеру (до 200 000 записей, до 5 000 проверок в минуту); выводить список последних 50 проходов в порядке убывания времени; формировать отчёт с посетителями, отсортированными по фамилии; обрабатывать очередь заявок на разовый пропуск в порядке поступления; поддерживать отмену последних действий оператора.

Задания для обучающихся: 1. Для каждого требования определите профиль операций (поиск, вставка, удаление, обход, упорядочивание). 2. Предложите структуру данных для каждого требования и обоснуйте выбор оценкой сложности. 3. Оцените суммарный расход памяти предложенного решения. 4. Укажите, какие требования конфликтуют между собой, и предложите компромисс. 5. Опишите, как изменится решение при росте числа записей до 20 млн.

Форма отчётности: письменное решение кейса с устной защитой на практическом занятии.

**Кейс № 2 «Алгоритм на графе» (к теме 1.9)**

Цель: сформировать навык сведения прикладной задачи к графовой модели и выбора алгоритма её решения.

Фабула: в корпоративной сети 60 узлов, соединённых каналами с различной пропускной способностью и задержкой.

Требуется определить маршрут с минимальной суммарной задержкой между двумя узлами, выявить узлы, отказ которых разрывает сеть, и определить, достижим ли каждый узел из центрального.

Задания для обучающихся: 1. Постройте графовую модель: определите тип графа, вершины, рёбра, веса. 2. Обоснуйте выбор представления графа с учётом его плотности. 3. Выберите и обоснуйте алгоритм для каждой из трёх подзадач. 4. Оцените сложность выбранных алгоритмов для заданной размерности. 5. Реализуйте решение первой подзадачи и приведите результат на тестовых данных. 6. Укажите, как изменится решение при появлении рёбер с отрицательным весом.

Форма отчётности: письменное решение кейса, программный код, устная защита.

**Кейс № 3 «Библиотека обработки заявок» (к темам 1.10—1.11)**

Цель: сформировать навыки объектно-ориентированного проектирования и разработки с использованием библиотек.

Фабула: сервисной компании требуется библиотека обработки заявок клиентов. Заявка имеет тип (инцидент, консультация, доработка), приоритет, срок реакции и исполнителя. Правила расчёта срока реакции различаются по типам, набор типов будет расширяться. Требуется поддержать выгрузку заявок в разных форматах и уведомление исполнителя различными способами. Задания для обучающихся: 1. Спроектируйте иерархию классов, обоснуйте выбор между наследованием и композицией. 2. Предложите решение, при котором добавление нового типа заявки не требует изменения существующего кода; укажите, какой принцип SOLID при этом реализуется. 3. Реализуйте библиотеку с покрытием ключевых сценариев тестами. 4. Обоснуйте выбор внешних библиотек и оцените риски зависимости от них. 5. Постройте диаграмму классов итогового решения.

Форма отчётности: программный код, диаграмма классов, письменное обоснование, устная защита.

Кейс № 4 «Медленный отчёт» (к темам 1.12—1.13)

Цель: сформировать навыки диагностики и устранения проблем производительности.

Фабула: формирование ежемесячного отчёта, ранее занимавшее 40 секунд, после роста объёма данных занимает 25 минут.

Код содержит вложенные циклы по двум коллекциям, повторное чтение файла в цикле и конкатенацию строк в цикле. Тесты отсутствуют.

Задания для обучающихся: 1. Составьте план диагностики и обоснуйте порядок действий. 2. Определите, какие изменения следует внести до оптимизации, чтобы не нарушить корректность. 3. Выполните профилирование и укажите три наиболее затратных участка. 4. Предложите и реализуйте оптимизацию, указав изменение асимптотической сложности. 5. Подтвердите результат замерами и объясните, почему устранение одного узкого места может выявить следующее.

Форма отчётности: письменное решение кейса, протокол замеров, устная защита.

Кейс № 5 «Настольное приложение для склада» (к теме 1.14)

Цель: сформировать навыки создания законченного приложения с пользовательским интерфейсом и воспроизводимым окружением.

Фабула: небольшой организации требуется приложение учёта товаров: добавление и редактирование позиций, списание, поиск, формирование отчёта об остатках. Пользователи — сотрудники без ИТ-подготовки. Приложение должно работать на отечественной операционной системе без доступа в интернет.

Задания для обучающихся: 1. Спроектируйте архитектуру приложения с разделением слоёв данных, логики и представления. 2. Обоснуйте выбор структур данных для хранения и поиска позиций. 3. Реализуйте базовый функционал с сохранением данных в файл. 4. Разработайте интерфейс с валидацией ввода и понятными сообщениями об ошибках. 5. Обеспечьте воспроизводимое развёртывание в закрытом контуре. 6. Покройте тестами слой бизнес-логики. 7. Опишите ограничения решения и направления его развития.

Форма отчётности: программный код, документация, устная защита с демонстрацией.

Примерный перечень вопросов к экзамену:

1. Понятие алгоритма и его свойства. Способы описания алгоритмов.
2. Базовые алгоритмические конструкции и принципы структурного программирования.
3. Требования ГОСТ 19.701-90 к оформлению блок-схем.
4. Классификация языков программирования. Компиляция и интерпретация.
5. Типы данных: статическая и динамическая типизация, строгая и слабая типизация.
6. Управляющие конструкции и их отображение в структурном программировании.
7. Функции: параметры, возвращаемые значения, области видимости, передача по значению и по ссылке.
8. Рекурсия: устройство стека вызовов, глубина рекурсии, риски, преобразование в итерацию.
9. Модульность программы: принципы декомпозиции, связность и сцепление.
10. Временная и ёмкостная сложность алгоритма. Асимптотические обозначения  $O$ ,  $\Omega$ ,  $\Theta$ .
11. Худший, средний и лучший случай: как и зачем они различаются.
12. Амортизированная сложность на примере динамического массива.
13. Массив и связный список: устройство, сравнение стоимости операций.
14. Стек: определение, операции, области применения.
15. Очередь и дек: определение, операции, области применения.
16. Коллекции языка высокого уровня: список, словарь, множество, кортеж; сложность операций и критерии выбора.
17. Сортировка вставками и пузырьком: принцип, сложность, применимость.
18. Сортировка слиянием: принцип, сложность, требования к памяти.
19. Быстрая сортировка: принцип, выбор опорного элемента, худший случай.
20. Поразрядная сортировка: условия применимости.
21. Устойчивость сортировки: определение и практическое значение.
22. Линейный и бинарный поиск: условия применимости, сложность.
23. Двоичное дерево поиска: структура, операции, вырожденный случай.
24. Сбалансированные деревья: назначение балансировки.
25. Куча и очередь с приоритетом: устройство и применение.
26. Хэш-таблица: хэш-функция, коллизии, методы разрешения коллизий.
27. Коэффициент заполнения хэш-таблицы и его влияние на производительность.
28. Способы представления графа и их сравнение.
29. Обход графа в глубину: алгоритм, сложность, применение.
30. Обход графа в ширину: алгоритм, сложность, применение.
31. Алгоритм Дейкстры: принцип работы и ограничения.
32. Динамическое программирование: признаки применимости, мемоизация и табличный подход.
33. Жадные алгоритмы и условия их корректности.
34. Критерии выбора алгоритма и структуры данных под прикладную задачу.
35. Соотношение теоретической оценки сложности и реального времени выполнения.

36. Файловый ввод-вывод. Форматы CSV и JSON: назначение и особенности обработки.
37. Обработка исключений: механизм, иерархия исключений, антипаттерны.
38. Парадигмы программирования: сопоставление процедурной, объектно-ориентированной и функциональной.
39. Класс и объект. Инкапсуляция и сокрытие реализации.
40. Наследование: назначение, риски, альтернативы.
41. Полиморфизм: формы проявления и практическое значение.
42. Абстрактные классы и интерфейсы: различия и применение.
43. Принципы SOLID: содержание и примеры нарушения.
44. Композиция и делегирование как альтернатива наследованию.
45. Функции высшего порядка, замыкания, неизменяемые данные.
46. Уровни тестирования: модульное, интеграционное, системное, приёмочное.
47. Техники тест-дизайна: классы эквивалентности, граничные значения, таблицы решений.
48. Модульное тестирование: структура теста, изоляция зависимостей, покрытие кода.
49. Тестирование алгоритмов: граничные и вырожденные наборы данных.
50. Отладка программ: точки останова, наблюдение состояния, логирование.
51. Признаки низкого качества кода и приёмы рефакторинга.
52. Профилирование и оптимизация: порядок действий и типовые ошибки.
53. Стиль оформления и документирование кода; назначение соглашений о кодировании.
54. Системы контроля версий: модель Git, ветвление, слияние, разрешение конфликтов.
55. Управление зависимостями и воспроизводимость сборки.
56. Изолированные окружения: назначение и способы создания.
57. Пользовательские интерфейсы: консольный, настольный, веб; критерии выбора.
58. Принципы UI/UX и валидация пользовательского ввода.
59. Программные средства разработки отечественного производства: назначение Единого реестра российского ПО, критерии выбора инструментов.
60. Внешние зависимости проекта: риски и способы их снижения.
61. Жизненный цикл программы от постановки задачи до сопровождения.

## 5.2. Темы письменных работ

не предусмотрены

## 5.3. Фонд оценочных средств

Тестовые задания:

### РАЗДЕЛ 1. ВОПРОСЫ НА СООТВЕТСТВИЕ (№ 1—9)

Инструкция: к каждому элементу из левого столбца подберите ОДИН соответствующий элемент из правого столбца. Ответ запишите в виде сочетания цифр и букв (например: 1-А, 2-Б, 3-В, 4-Г). Каждая буква используется только один раз.

ОПК-7.1. Вопрос № 1. Установите соответствие между нотацией описания алгоритма и её характеристикой:

1. Блок-схема — 2. Псевдокод — 3. UML-диаграмма деятельности — 4. Исходный код (провокация).

Варианты: А. Графическая нотация, регламентированная ГОСТ 19.701-90; Б. Полуформальная текстовая запись, независимая от языка программирования; В. Графическая нотация моделирования потока работ, допускающая параллельные ветви; Г. Реализация алгоритма средствами конкретного языка.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-7.1. Вопрос № 2. Установите соответствие между алгоритмом сортировки и его средней временной сложностью:

1. Сортировка пузырьком — 2. Сортировка слиянием — 3. Поразрядная сортировка — 4. Сортировка выбором.

Варианты: А.  $O(n \log n)$ ; Б.  $O(n^2)$  с ранним выходом на отсортированных данных; В.  $O(k \cdot n)$ ; Г.  $O(n^2)$  без зависимости от исходного порядка.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-7.2. Вопрос № 3. Установите соответствие между структурой данных и операцией, которую она выполняет за  $O(1)$  в среднем:

1. Хэш-таблица — 2. Стек — 3. Динамический массив — 4. Двусвязный список.

Варианты: А. Извлечение последнего добавленного элемента; Б. Поиск по ключу; В. Доступ по индексу; Г. Удаление элемента при известной ссылке на узел.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-7.2. Вопрос № 4. Установите соответствие между задачей и подходящим алгоритмом:

1. Кратчайший путь во взвешенном графе с неотрицательными весами — 2. Проверка связности графа — 3. Поиск в отсортированном массиве — 4. Кратчайший путь между всеми парами вершин.

Варианты: А. Обход в глубину; Б. Алгоритм Дейкстры; В. Алгоритм Флойда — Уоршелла; Г. Бинарный поиск.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-7.2. Вопрос № 5. Установите соответствие между парадигмой программирования и её ключевым понятием:

1. Процедурная — 2. Объектно-ориентированная — 3. Функциональная — 4. Логическая (провокация).

Варианты: А. Класс и объект; Б. Подпрограмма и последовательность действий; В. Чистая функция и неизменяемые данные; Г. Факт, правило и вывод.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-7.1. Вопрос № 6. Установите соответствие между принципом ООП и его содержанием:

1. Инкапсуляция — 2. Наследование — 3. Полиморфизм — 4. Абстракция.

Варианты: А. Передача свойств и поведения от базового типа к производному; Б. Скрытие внутреннего состояния и предоставление контролируемого интерфейса; В. Единообразная работа с объектами разных типов через общий интерфейс; Г. Выделение существенных характеристик объекта с отбрасыванием второстепенных.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-5.2. Вопрос № 7. Установите соответствие между уровнем тестирования и его назначением:

1. Модульное — 2. Интеграционное — 3. Системное — 4. Приёмочное.

Варианты: А. Проверка взаимодействия компонентов между собой; Б. Проверка отдельной функции или класса в изоляции; В. Проверка соответствия системы требованиям заказчика; Г. Проверка системы целиком в среде, приближенной к эксплуатационной.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-7.2. Вопрос № 8. Установите соответствие между командой системы контроля версий Git и её действием:

1. commit — 2. branch — 3. merge — 4. clone.

Варианты: А. Создание новой ветки разработки; Б. Фиксация изменений в истории репозитория; В. Создание локальной копии удалённого репозитория; Г. Объединение изменений одной ветки с другой.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

ОПК-5.2. Вопрос № 9. Установите соответствие между приёмом рефакторинга и решаемой проблемой:

1. Выделение функции — 2. Замена «магического числа» именованной константой — 3. Замена условного оператора полиморфизмом — 4. Переименование переменной.

Варианты: А. Непонятное назначение литерала в коде; Б. Слишком длинный метод, выполняющий несколько задач; В. Неинформативное имя, затрудняющее чтение кода; Г. Разрастающийся разбор случаев по типу объекта.

Ответ: 1 \_\_, 2 \_\_, 3 \_\_, 4 \_\_

## РАЗДЕЛ 2. ВОПРОСЫ С ОДНИМ ВЕРНЫМ ОТВЕТОМ (№ 10—20)

Инструкция: прочитайте текст, выберите правильный вариант ответа и запишите только букву выбранного варианта (А, Б, В или Г).

ОПК-7.2. Вопрос № 10. Какова сложность бинарного поиска в отсортированном массиве из  $n$  элементов?

А)  $O(1)$ ; Б)  $O(\log n)$ ; В)  $O(n)$ ; Г)  $O(n \log n)$ .

Ответ:

ОПК-7.2. Вопрос № 11. Какая структура данных реализует принцип «последним пришёл — первым вышел»?

А) Очередь; Б) Стек; В) Куча; Г) Хэш-таблица.

Ответ:

ОПК-7.2. Вопрос № 12. Какой алгоритм обхода графа используется для поиска кратчайшего пути в невзвешенном графе?

А) Обход в глубину; Б) Обход в ширину; В) Алгоритм Флойда — Уоршелла; Г) Быстрая сортировка.

Ответ:

ОПК-5.2. Вопрос № 13. Какова временная сложность быстрой сортировки в худшем случае?

А)  $O(n)$ ; Б)  $O(n \log n)$ ; В)  $O(n^2)$ ; Г)  $O(\log n)$ .

Ответ:

ОПК-5.2. Вопрос № 14. Что происходит при возникновении коллизии в хэш-таблице?

А) Элемент не может быть добавлен; Б) Два различных ключа отображаются в одну ячейку и обрабатываются выбранным методом разрешения коллизий; В) Таблица автоматически удаляет старый элемент; Г) Хэш-функция заменяется на новую.

Ответ:

ОПК-7.1. Вопрос № 15. Что произойдёт при возникновении необработанного исключения в программе?

А) Программа продолжит выполнение со следующей строки; Б) Выполнение прервётся с выводом сообщения об ошибке; В) Исключение будет автоматически исправлено средой выполнения; Г) Программа перезапустится.

Ответ:

ОПК-7.2. Вопрос № 16. Какое отношение между классами корректно моделируется наследованием?

А) «Содержит»; Б) «Использует»; В) «Является»; Г) «Создаёт».

Ответ:

ОПК-5.2. Вопрос № 17. Что показывает метрика покрытия кода тестами?

А) Долю кода, выполненную при прогоне тестов; Б) Долю тестов, завершившихся успешно; В) Количество найденных дефектов; Г) Отсутствие ошибок в программе.

Ответ:

ОПК-5.2. Вопрос № 18. Для чего предназначена система контроля версий?

А) Для компиляции исходного кода; Б) Для хранения истории изменений и организации совместной работы над кодом; В) Для автоматического исправления ошибок; Г) Для измерения производительности программы.

Ответ:

ОПК-5.2. Вопрос № 19. Какова основная цель использования изолированного виртуального окружения проекта?

А) Ускорение работы программы; Б) Разграничение прав доступа пользователей; В) Изоляция набора и версий зависимостей проекта от системных и других проектов; Г) Сжатие исходного кода.

Ответ:

ОПК-5.2. Вопрос № 20. Что представляет собой Единый реестр российского программного обеспечения?

А) Перечень запрещённого к использованию ПО; Б) Перечень программ и баз данных, признанных происходящими из Российской Федерации, применяемый в том числе при закупках для государственных нужд; В) Каталог свободных лицензий; Г) Реестр разработчиков программного обеспечения.

Ответ:

### РАЗДЕЛ 3. ВОПРОСЫ С ОДНИМ АРГУМЕНТИРОВАННЫМ ОТВЕТОМ (№ 21—28)

Инструкция: прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор.

ОПК-7.1. Вопрос № 21. Какое утверждение о свойствах алгоритма верно?

А) Алгоритм может не завершаться на корректных входных данных; Б) Алгоритм должен обладать конечностью, детерминированностью, результативностью и массовостью; В) Алгоритм всегда описывается только на языке программирования; Г) Алгоритм не требует определения входных данных.

Ответ:

ОПК-5.2. Вопрос № 22. Какая сортировка предпочтительна для почти отсортированного массива небольшого размера?

А) Быстрая; Б) Сортировка вставками; В) Сортировка выбором; Г) Поразрядная.

Ответ:

ОПК-7.2. Вопрос № 23. Что следует выбрать для хранения данных, если требуется частый поиск по ключу и порядок элементов не важен?

А) Связный список; Б) Хэш-таблица; В) Массив без сортировки; Г) Стек.

Ответ:

ОПК-7.2. Вопрос № 24. Почему наивная рекурсивная реализация чисел Фибоначчи неэффективна?

А) Рекурсия не поддерживается компиляторами; Б) Из-за многократного повторного вычисления одних и тех же подзадач сложность становится экспоненциальной; В) Из-за переполнения типа данных; Г) Из-за отсутствия базового случая.

Ответ:

ОПК-5.2. Вопрос № 25. Что необходимо сделать в первую очередь перед рефакторингом кода, не покрытого тестами?

А) Немедленно начать переработку структуры; Б) Покрыть текущее поведение тестами; В) Переписать модуль с нуля; Г) Оптимизировать самые медленные участки.

Ответ:

ОПК-7.2. Вопрос № 26. Какое решение предпочтительно, если производный класс не может корректно выполнять контракт базового?

А) Переопределить метод и выбрасывать исключение; Б) Отказаться от наследования и применить композицию; В) Сделать метод базового класса пустым; Г) Игнорировать проблему.

Ответ:

ОПК-5.2. Вопрос № 27. Что означает 100 % покрытие кода тестами?

А) Программа не содержит ошибок; Б) Все строки кода были выполнены хотя бы раз при прогоне тестов, что не гарантирует отсутствия дефектов; В) Все возможные комбинации входных данных проверены; Г) Тесты не требуют дальнейшей поддержки.

Ответ:

ОПК-7.2. Вопрос № 28. Почему не рекомендуется перехватывать все исключения одним обработчиком без анализа типа?

А) Это замедляет программу; Б) Такой обработчик скрывает непредвиденные ошибки и затрудняет диагностику, маскируя дефекты; В) Это запрещено синтаксисом языка; Г) Это увеличивает объём исполняемого файла.

Ответ:

### РАЗДЕЛ 4. ВОПРОСЫ С НЕСКОЛЬКИМИ ВЕРНЫМИ ОТВЕТАМИ (№ 29—36)

Инструкция: выберите не менее двух правильных вариантов и запишите только буквы выбранных вариантов.

ОПК-7.1. Вопрос № 29. Какие элементы обязательно указываются при постановке задачи для построения алгоритма?

А) Входные данные; Б) Выходные данные; В) Условия завершения; Г) Марка используемого процессора.

Ответ:

ОПК-7.1. Вопрос № 30. Какие структуры данных обеспечивают доступ к произвольному элементу по индексу за  $O(1)$ ?

А) Массив; Б) Динамический массив; В) Односвязный список; Г) Двусвязный список.

Ответ:

ОПК-5.2. Вопрос № 31. Какие сортировки в среднем работают за  $O(n \log n)$ ?

А) Сортировка слиянием; Б) Быстрая сортировка; В) Пирамидальная сортировка; Г) Сортировка выбором

Ответ:

ОПК-7.2. Вопрос № 32. Какие задачи решаются методом динамического программирования?

А) Задача о рюкзаке; Б) Поиск наибольшей общей подпоследовательности; В) Определение чётности числа; Г) Задача о размене монет.

Ответ:

ОПК-7.1. Вопрос № 33. Какие принципы входят в состав SOLID?

А) Принцип единственной ответственности; Б) Принцип открытости/закрытости; В) Принцип подстановки Лисков; Г) Принцип максимального наследования.

Ответ:

ОПК-5.2. Вопрос № 34. Какие техники относятся к тест-дизайну?

А) Классы эквивалентности; Б) Анализ граничных значений; В) Таблицы решений; Г) Профилирование кода.

Ответ:

ОПК-5.2. Вопрос № 35. Какие признаки указывают на необходимость рефакторинга?

А) Дублирование кода; Б) Чрезмерно длинные функции; В) «Магические числа» в коде; Г) Наличие модульных тестов.

Ответ:

ОПК-7.2. Вопрос № 36. Какие требования относятся к качественному пользовательскому интерфейсу?

А) Валидация вводимых данных; Б) Понятные сообщения об ошибках; В) Минимальное число шагов до результата; Г) Максимальное количество элементов на экране.

Ответ:

## РАЗДЕЛ 5. ВОПРОСЫ С ОТКРЫТЫМ РАЗВЕРНУТЫМ ОТВЕТОМ (№ 37—44)

Инструкция: прочитайте текст и запишите развернутый обоснованный ответ.

ОПК-7.1. Вопрос № 37. Перечислите этапы разработки алгоритма от постановки задачи до тестирования и укажите результат каждого этапа.

Ответ:

ОПК-7.2. Вопрос № 38. Объясните, что показывает асимптотическая оценка сложности и почему алгоритм с меньшей асимптотикой может проигрывать на малых объёмах данных.

Ответ:

ОПК-7.2. Вопрос № 39. Сравните хэш-таблицу и двоичное дерево поиска по критериям: сложность операций, упорядоченность, устойчивость к неудачным данным, расход памяти.

Ответ:

ОПК-7.2. Вопрос № 40. Опишите различия между обходом графа в глубину и в ширину и укажите, для каких задач применяется каждый.

Ответ:

ОПК-5.2. Вопрос № 41. Объясните, что такое связность и сцепление модулей и почему стремятся к высокой связности при низком сцеплении.

Ответ:

ОПК-5.2. Вопрос № 42. Опишите типовой порядок командной работы с репозиторием Git при разработке новой функции.

Ответ:

ОПК-5.2. Вопрос № 43. Перечислите условия, обеспечивающие воспроизводимость сборки проекта на другой машине.

Ответ:

ОПК-7.2. Вопрос № 44. Опишите порядок проектирования пользовательского интерфейса прикладной программы.

Ответ:

## РАЗДЕЛ 6. ВОПРОСЫ НА УСТАНОВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ (№ 45—48)

Инструкция: установите правильную последовательность действий и запишите аргументы, обосновывающие выбранный порядок.

ОПК-7.1. Вопрос № 45. Установите последовательность этапов решения прикладной задачи средствами алгоритмизации.

А) Реализация на языке программирования; Б) Формализация постановки задачи; В) Тестирование и анализ результатов; Г) Выбор структуры данных и метода решения; Д) Оценка вычислительной сложности.

Ответ:

ОПК-7.2. Вопрос № 46. Установите последовательность шагов сортировки слиянием.

А) Слияние двух отсортированных половин в один упорядоченный массив; Б) Рекурсивная сортировка каждой половины; В) Разделение массива на две примерно равные части; Г) Проверка базового случая (длина массива не более одного элемента).

Ответ:

ОПК-5.2. Вопрос № 47. Установите последовательность этапов разработки модульных тестов.

А) Реализация тестов в выбранном фреймворке; Б) Выделение классов эквивалентности и граничных значений; В) Анализ требований к тестируемому модулю; Г) Анализ отчёта о покрытии и дополнение набора тестов; Д) Составление таблицы тест-кейсов с ожидаемым результатом.

Ответ:

ОПК-5.2. Вопрос № 48. Установите последовательность действий при оптимизации медленно работающей программы.

А) Внесение изменений в критический участок; Б) Профилирование и определение узких мест; В) Покрытие текущего поведения тестами; Г) Повторный замер и сравнение результатов; Д) Постановка измеримой цели оптимизации.

Ответ:

#### 5.4. Перечень видов оценочных средств и критерии оценивания

Контроль результатов освоения дисциплины является формой управления качеством образования.

Контроль результатов освоения дисциплины осуществляется в виде текущего контроля и промежуточной аттестации (экзамена).

Система текущего контроля включает:

1. контроль работы обучающихся на лекционных и практических занятиях;
2. контроль участия в совместной работе группы;
3. контроль выполнения индивидуальных заданий;
4. контроль выполнения обучающимися заданий для самостоятельной работы.

Промежуточная аттестация осуществляется в виде экзамена, который проводится в устной либо письменной форме.

Задачи промежуточной аттестации:

1. определить общий уровень освоения дисциплины в целом;
2. определить уровень основных знаний по каждой теме курса;
3. объективизировать результат контроля, минимизировать возможную субъективность преподавателя.

Критерии и рубрики оценивания (элемент контроля / критерий / индикаторы / уровни 0—3—5 / комментарий к уровню «5»)

Задание № 1 «Первая программа и работа с отладчиком» — корректность реализации и владение инструментами / ОПК-1.2, ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Программа корректно работает на всех сценариях, включая некорректный ввод; история репозитория содержит осмысленные коммиты, продемонстрированы ветвление и слияние; отладка выполнена с пояснением состояния переменных; оформлена документация запуска.

Задание № 2 «Формализация алгоритма» — корректность формального описания алгоритма / ОПК-7.1 / 0 — 3 — 5 / Выделены вход, выход, предусловия и постусловия; блок-схема соответствует ГОСТ 19.701-90; псевдокод и диаграмма деятельности согласованы между собой; определены все ветви алгоритма и подобраны тестовые наборы на каждую ветвь.

Задание № 3 «Модульная программа» — качество декомпозиции и обработки данных / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Модули разделены по зонам ответственности, ввод-вывод отделён от логики; реализована проверка корректности данных и обработка исключений; все показатели рассчитаны верно; выбор коллекций обоснован оценкой сложности; результат корректно выгружается в JSON.

Задание № 4 «Оценка сложности» — обоснованность асимптотической оценки / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Для каждого фрагмента верно определены временная и ёмкостная сложность в худшем и среднем случае с указанием доминирующей операции; выполнено корректное упорядочение фрагментов; расчётом подтверждён порог практической значимости различий; предложены обоснованные способы оптимизации.

Задание № 5 «Реализация и тестирование алгоритмов» — корректность реализации и полнота тестирования / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Все алгоритмы реализованы и проходят тесты; покрыты граничные и вырожденные наборы данных; результаты замеров оформлены таблицей и графиком; расхождение теоретической оценки и измеренного времени объяснено; устойчивость сортировок подтверждена тестом.

Задание № 6 «Модульные тесты» — полнота и качество тестирования / ОПК-5.2 / 0 — 3 — 5 / Выделены классы эквивалентности и граничные значения; таблица тест-кейсов полна и согласована с кодом; реализовано не менее 15 тестов, включая негативные; отчёт о покрытии проанализирован, непокрытые ветви объяснены; продемонстрировано обнаружение внесённого дефекта.

Задание № 7 «Рефакторинг и профилирование» — повышение качества и производительности кода / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Выявлены все признаки низкого качества кода; применено не менее пяти обоснованных приёмов рефакторинга; сохранение поведения подтверждено тестами; приведены замеры до и после; корректно оценено изменение асимптотической сложности.

Задание № 8 «Окружение проекта» — воспроизводимость конфигурации / ОПК-5.2, ОПК-7.1 / 0 — 3 — 5 / Зависимости зафиксированы с указанием версий; параметры среды вынесены из кода; развёртывание «с нуля» выполнено успешно на отечественной операционной системе; инструкция полна и проверена сторонним исполнителем; описаны риски обновления зависимостей.

Задание № 9 «Пользовательский интерфейс» — качество проектирования и реализации интерфейса / ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Описаны сценарии использования; интерфейс реализован и соответствует эскизам; ввод валидируется, сообщения об ошибках понятны неподготовленному пользователю; решение оценено по заявленным критериям с аргументацией.

Кейс № 1 «Выбор структуры данных для сервиса» — обоснованность выбора структуры данных / ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Для каждого требования определён профиль операций и предложена структура данных с оценкой сложности; рассчитан расход памяти; выявлены конфликтующие требования и предложен компромисс; описано масштабирование решения при росте объёма данных.

Кейс № 2 «Алгоритм на графе» — корректность графовой модели и выбора алгоритма / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Построена корректная графовая модель с обоснованием типа графа и способа представления; для каждой подзадачи выбран обоснован алгоритм с оценкой сложности; реализация даёт верный результат на тестовых данных; корректно описано влияние отрицательных весов.

Кейс № 3 «Библиотека обработки заявок» — объектно-ориентированное проектирование / ОПК-1.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Иерархия классов корректна, выбор между наследованием и композицией обоснован; расширение новым типом заявки не требует изменения существующего кода со ссылкой на принцип открытости/закрытости; ключевые сценарии покрыты тестами; диаграмма классов соответствует реализации.

Кейс № 4 «Медленный отчёт» — диагностика и устранение проблем производительности / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / План диагностики обоснован; до оптимизации зафиксировано поведение тестами; определены три наиболее затратных участка по данным профилировщика; оптимизация выполнена с указанием изменения сложности; результат подтверждён замерами.

Кейс № 5 «Настольное приложение для склада» — разработка законченного приложения / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Архитектура разделена на слои; выбор структур данных обоснован; функционал реализован полностью; интерфейс пригоден для неподготовленного пользователя; развёртывание в закрытом контуре воспроизводимо; бизнес-логика покрыта тестами; ограничения решения описаны корректно.

Семинары № 1, 4 — участие в обсуждении и аргументация проектных решений / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Обучающийся сопоставляет парадигмы и подходы к проектированию по объективным критериям, аргументирует выбор, корректно оперирует принципами SOLID и отвечает на встречные вопросы.

Семинары № 2, 3 — участие в обсуждении и аргументация алгоритмического решения / ОПК-5.2, ОПК-7.1, ОПК-7.2 / 0 — 3 — 5 / Обучающийся сопоставляет структуры данных и алгоритмы по стоимости операций, аргументирует выбор ссылкой на асимптотические оценки и профиль нагрузки, корректно отвечает на встречные вопросы.

Семинар № 5 — обоснование выбора инструментальной среды / ОПК-5.2 / 0 — 3 — 5 / Средства разработки сопоставлены по функциональным критериям; учтены требования к отечественному ПО и Единому реестру российского ПО; окружение развёрнуто и воспроизводимость сборки подтверждена.

Тест — корректность ответов на вопросы по алгоритмам, структурам данных и программированию / ОПК-1.2, ОПК-5.2, ОПК-7.1, ОПК-7.2 / % → балл /  $\geq 90\%$  — 5; 75—89 % — 4; 60—74 % — 3;  $< 60\%$  — 2.

Критерии экзаменационной оценки: «отлично» — исчерпывающий ответ на теоретический вопрос, безошибочно выполненное практическое задание с обоснованием алгоритмических и проектных решений, все работы семестра сданы; «хорошо» — ответ с несущественными неточностями, практическое задание выполнено с незначительными замечаниями; «удовлетворительно» — фрагментарный ответ, практическое задание выполнено частично, но продемонстрировано понимание базовых алгоритмов и языковых конструкций; «неудовлетворительно» — незнание базовых понятий либо невыполнение практического задания.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 6.1. Рекомендуемая литература

#### Основная литература

Гаврилов, М. В. Информатика и информационные технологии : учебник для вузов / М. В. Гаврилов, В. А. Климов. — 7-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 307 с. — (Высшее образование). — ISBN 978-5-534-19791-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/600409>.

Дьячков, В. П. Информатика. Устройство персонального компьютера : учебник и практикум для вузов / В. П. Дьячков. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 128 с. — (Высшее образование). — ISBN 978-5-534-19227-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590341>.

Зараменских, Е. П. Проектирование информационных систем : учебник и практикум для вузов / Е. П. Зараменских. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 119 с. — (Высшее образование). — ISBN 978-5-534-21418-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/571331>.

Зыков С. В. Программирование : учебник и практикум для вузов. — Москва : Издательство Юрайт, 2023. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — URL: <https://urait.ru/bcode/489754>.

Кудрявцева И. А., Швецкий М. В. Программирование: комбинаторная логика : учебное пособие для СПО. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 524 с. — (Профессиональное образование). — ISBN 978-5-534-15128-2. — URL: <https://urait.ru/bcode/542030>.

Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561885>.

Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2025. — 176 с. — (Высшее образование). — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567946>.

#### Дополнительная литература

Федоров Д. Ю. Программирование на Python : учебное пособие для вузов. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Высшее образование). — ISBN 978-5-534-19666-5. — URL: <https://urait.ru/bcode/556864>.

Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования : учебное пособие для вузов. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — URL: <https://urait.ru/bcode/536903>.

Трофимов В. В., Павловская Т. А. Алгоритмизация и программирование : учебник для вузов. — 4-е изд. — Москва : Издательство Юрайт, 2024. — 118 с. — (Высшее образование). — ISBN 978-5-534-17497-7. — URL: <https://urait.ru/bcode/538039>.

Тузовский А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов. — Москва : Издательство Юрайт. — (Высшее образование). — ISBN 978-5-534-00849-4. — URL: <https://urait.ru/bcode/470223>.

| <b>6.2. Электронные учебные издания и электронные образовательные ресурсы</b>                                          |                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Э1                                                                                                                     | Сетевые ресурсы Государственного автономного учреждения культуры Свердловской области «Свердловская областная универсальная научная библиотека им. В.Г. Белинского». Режим доступа: <a href="https://book.uraic.ru/el_library/resursy_dostup">https://book.uraic.ru/el_library/resursy_dostup</a> |
| Э2                                                                                                                     | Образовательная платформа Юрайт. Режим доступа: <a href="https://urait.ru/">https://urait.ru/</a>                                                                                                                                                                                                 |
| Э3                                                                                                                     | Научная библиотека Cyberleninka. Режим доступа: <a href="https://cyberleninka.ru/">https://cyberleninka.ru/</a>                                                                                                                                                                                   |
| Э4                                                                                                                     | Опубликование стандартов, классификаторов, сводов правил. Режим доступа: <a href="https://protect.gost.ru">https://protect.gost.ru</a>                                                                                                                                                            |
| <b>6.3.1 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства</b> |                                                                                                                                                                                                                                                                                                   |
| 6.3.1.1                                                                                                                | Операционная система Astra Linux                                                                                                                                                                                                                                                                  |
| 6.3.1.2                                                                                                                | Офисная система LibreOffice                                                                                                                                                                                                                                                                       |
| 6.3.1.3                                                                                                                | Интерпретатор Python 3, компилятор GCC (свободно распространяемое ПО)                                                                                                                                                                                                                             |
| 6.3.1.4                                                                                                                | Среда разработки со свободной лицензией (VS Code / OSS-сборка), система контроля версий Git, фреймворк модульного тестирования, профилировщик                                                                                                                                                     |
| 6.3.1.5                                                                                                                | Контур.Класс                                                                                                                                                                                                                                                                                      |
| 6.3.1.6                                                                                                                | Контур.Толк                                                                                                                                                                                                                                                                                       |
| 6.3.1.7                                                                                                                | 1С:Университет ПРОФ                                                                                                                                                                                                                                                                               |
| 6.3.1.8                                                                                                                | Среда электронного обучения 3LK (русский Moodle)                                                                                                                                                                                                                                                  |

| <b>7. МТО (оборудование и технические средства обучения)</b> |                                                                                                                                                                                                                        |
|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7.1                                                          | Учебные аудитории для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенные оборудованием и техническими средствами обучения:                                                                |
| 7.2                                                          | Адрес (местоположение) учебных аудиторий Перечень оборудования и технических средств обучения                                                                                                                          |
| 7.3                                                          | 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 3 площадью 66,2 м² в помещении 501                                                                  |
| 7.4                                                          | Стол – 13 шт.                                                                                                                                                                                                          |
| 7.5                                                          | Стул – 25 шт.                                                                                                                                                                                                          |
| 7.6                                                          | ПК (с возможностью подключения к сети «Интернет») – 1 шт.                                                                                                                                                              |
| 7.7                                                          | Телевизор (75") – 1 шт.                                                                                                                                                                                                |
| 7.8                                                          | Беспроводная система презентаций – 1 шт.                                                                                                                                                                               |
| 7.9                                                          | Спикерфон – 1 шт.                                                                                                                                                                                                      |
| 7.10                                                         | ПК (ноутбук) – 12 шт.                                                                                                                                                                                                  |
| 7.11                                                         | 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 19 площадью 31,7 м² в помещении 501                                                                 |
| 7.12                                                         | Стол – 7 шт.                                                                                                                                                                                                           |
| 7.13                                                         | Стул – 13 шт.                                                                                                                                                                                                          |
| 7.14                                                         | ПК (с возможностью подключения к сети «Интернет») – 1 шт.                                                                                                                                                              |
| 7.15                                                         | Магнитно-маркерная доска – 1 шт.                                                                                                                                                                                       |
| 7.16                                                         | Телевизор (75") – 1 шт.                                                                                                                                                                                                |
| 7.17                                                         | 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 21 площадью 35,6 м² в помещении 501                                                                 |
| 7.18                                                         | Стол – 7 шт.                                                                                                                                                                                                           |
| 7.19                                                         | Стул – 13 шт.                                                                                                                                                                                                          |
| 7.20                                                         | ПК (с возможностью подключения к сети «Интернет») – 1 шт.                                                                                                                                                              |
| 7.21                                                         | Магнитно-маркерная доска – 1 шт.                                                                                                                                                                                       |
| 7.22                                                         | Телевизор (75") – 1 шт.                                                                                                                                                                                                |
| 7.23                                                         |                                                                                                                                                                                                                        |
| 7.24                                                         | Помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета: |
| 7.25                                                         | Адрес (местоположение) помещения для самостоятельной работы обучающихся Перечень компьютерной техники и оборудования                                                                                                   |
| 7.26                                                         | 620014, Свердловская область, г.о. город Екатеринбург, г Екатеринбург, пр-кт Ленина, стр. 20а, этаж № 5, кабинет № 5 площадью 44,3 м² в помещении 501                                                                  |
| 7.27                                                         | ПК (ноутбук) – 9 шт.                                                                                                                                                                                                   |
| 7.28                                                         | Стол – 11 шт.                                                                                                                                                                                                          |
| 7.29                                                         | Стул – 11 шт.                                                                                                                                                                                                          |

## 8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для успешного освоения дисциплины рекомендуется ознакомиться с рабочей программой, целями и задачами дисциплины, её связями с другими дисциплинами образовательной программы, изучить основную и дополнительную литературу и перечень заданий.

Дисциплина объединяет алгоритмическую и языковую подготовку: теоретический материал об алгоритмах и структурах данных изучается параллельно с освоением языка программирования, поэтому каждый алгоритм рекомендуется не только разобрать на бумаге, но и реализовать, протестировать и измерить.

Программирование осваивается только практикой: рекомендуется писать код ежедневно, доводя каждое учебное задание до работоспособного и протестированного состояния, а не до состояния «в основном работает».

В ходе лекций следует вести конспектирование, обращая внимание на определения, синтаксические конструкции, оценки сложности, условия применимости алгоритмов и типовые ошибки.

При подготовке к практическим занятиям необходимо заранее изучить теоретический материал и подготовить черновой вариант решения; на занятии решение доводится до рабочего состояния и защищается.

Работа с алгоритмами требует систематичности: рекомендуется еженедельно решать задачи на изученные структуры данных, обязательно проверяя решения на граничных и вырожденных наборах данных.

Весь учебный код рекомендуется вести в системе контроля версий с осмысленными сообщениями коммитов — это формирует профессиональную привычку и позволяет преподавателю оценить ход работы, а не только итог.

При пропуске занятия следует самостоятельно изучить материал и предоставить преподавателю отчёт по пропущенной теме в часы индивидуальных консультаций.

При подготовке к текущей аттестации необходимо изучать материал по блокам тем, акцентируя внимание на определениях, оценках сложности и критериях выбора алгоритма и структуры данных.

При подготовке к экзамену необходимо повторить весь материал дисциплины, обязательно разобрав ранее допущенные ошибки, и быть готовым не только изложить теорию, но и написать и объяснить работающий фрагмент кода.