МИНОБРНАУКИ РОССИИ
Глазовский инженерно-экономический институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Ижевский государственный технический университет имени М.Т.Калашникова»
(ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 3DA273AA38C240BA869FA062FFFB332C
Владелец: Золотов Алексей Васильевич
Действителен: с 29.07.2026 до 22.10.2027

УТВЕРЖДАЮ
И.о. директора
_____ А.В. Золотов
11 марта 20 26 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

ОП.06 «Основы алгоритмизации и программирования»

09.02.11 Разработка и управление программным обеспечением

Квалификация **Программист**

Глазов 2026

Фонд оценочных средств разработан на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.11 "Разработка и управление программным обеспечением", утвержденного приказом Министерства просвещения Российской Федерации 24 февраля 2025 г. № 138 (зарегистрирован 31.03.2025 № 81696).

Организация разработчик: ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Разработчик: Горбушин Денис Шарибзянович, преподаватель СПО

Утверждено: Протокол Ученого совета филиала № 7, от 11 марта 2026 г.

Руководитель образовательной программы

_____ Т.А. Савельева

13 марта 2026 г.

.

Согласовано: Начальник отдела по учебно-методической работе

_____ И.Ф. Яковлева

13 марта 2026 г.

ПАСПОРТ
ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине «Основы алгоритмизации и программирования»

Результаты освоения дисциплины соотносятся с планируемыми результатами освоения образовательной программы, представленными в матрице компетенций выпускника.

В результате освоения дисциплины обучающийся должен:

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	– распознавать задачу и/или проблему в профессиональном и/или социальном контексте; – анализировать задачу и/или проблему и выделять её составные части; – определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; – составить план действия; определить необходимые ресурсы; – владеть актуальными методами работы в профессиональной и смежных сферах; – реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	– актуальный профессиональный и социальный контекст, в котором приходится работать и жить; – основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; – алгоритмы выполнения работ в профессиональной и смежных областях; – методы работы в профессиональной и смежных сферах; – структуру плана для решения задач; – порядок оценки результатов решения задач профессиональной деятельности	-
ОК 02. Осуществлять поиск, анализ и интерпретацию	– определять задачи для поиска информации, планировать процесс поиска, выбирать	– номенклатура информационных источников, применяемых в	-

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
информации, необходимой для выполнения задач профессиональной деятельности.	необходимые источники информации	профессиональной деятельности	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	– грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе	– особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.	
ОК 09. Использовать информационные технологии в профессиональной деятельности.	– применять средства информационных технологий для решения профессиональных задач; – использовать современное программное обеспечение	– современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности	-
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.	– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; – участвовать в диалогах на знакомые общие и профессиональные темы; – строить простые высказывания о себе и о своей профессиональной деятельности; кратко	– правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); – лексический минимум, относящийся к	

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
	обосновывать и объяснить свои действия (текущие и планируемые); – писать простые связные сообщения на знакомые или интересующие профессиональные темы	описанию предметов, средств и процессов профессиональной деятельности; – особенности произношения; правила чтения текстов профессиональной направленности	
ПК-1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	– Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. – Оформлять документацию на программные средства. – Оценка сложности алгоритма.	– Основные этапы разработки программного обеспечения. – Основные принципы технологии структурного и объектно-ориентированного программирования. – Актуальная нормативно-правовая база в области документирования алгоритмов.	– Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.	– Создавать программу по разработанному алгоритму как отдельный модуль. – Оформлять документацию на программные средства. – Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.	– Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектноориентированного программирования. – Знание API современных мобильных операционных систем	– Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля. – Разрабатывать мобильные приложения
ПК.1.3. Выполнять	– Выполнять отладку и тестирование программы	– Основные принципы	– Использовать инструментальн

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
отладку программных модулей с использованием специализированных программных средств.	на уровне модуля. – Оформлять документацию на программные средства. – Применять инструментальные средства отладки программного обеспечения.	отладки и тестирования программных продуктов. – Инструментарий отладки программных продуктов	ые средства на этапе отладки программного продукта. – Проводить тестирование программного модуля по определенному сценарию.
ПК 1.4. Выполнять тестирование программных модулей.	– Выполнять отладку и тестирование программы на уровне модуля. – Оформлять документацию на программные средства.	– Основные виды и принципы тестирования программных продуктов.	– Проводить тестирование программного модуля по определенному сценарию. – Использовать инструментальные средства на этапе тестирования программного продукта.
ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.	– Выполнять оптимизацию и рефакторинг программного кода. – Работать с системой контроля версий.	– Способы оптимизации и приемы рефакторинга. – Инструментальные средства анализа алгоритма. – Методы организации рефакторинга и оптимизации кода. – Принципы работы с системой контроля версий	– Анализировать алгоритмы, в том числе с применением инструментальных средств. – Осуществлять рефакторинг и оптимизацию программного кода.
ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	– Использовать выбранную систему контроля версий. – Анализировать проектную и техническую документацию. – Выполнять тестирование интеграции. – Организовывать	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного обеспечения.	– Разрабатывать тестовые наборы (пакеты) для программного модуля. – Разрабатывать тестовые сценарии программного средства.

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
	постобработку данных. – Использовать приемы работы в системах контроля версий. – Оценивать размер минимального набора тестов. – Разрабатывать тестовые пакеты и тестовые сценарии. – Выполнять ручное и автоматизированное тестирование программного модуля. – Выявлять ошибки в системных компонентах на основе спецификаций.	– Основные подходы к интегрированию программных модулей. – Основы верификации и аттестации программного обеспечения. – Методы и способы идентификации сбоев и ошибок при интеграции приложений. – Методы и схемы обработки исключительных ситуаций. – Основные методы и виды тестирования программных продуктов. – Приемы работы с инструментальными средствами тестирования и отладки. – Стандарты качества программной документации. – Основы организации инспектирования и верификации. – Встроенные и основные специализированные инструменты анализа качества программных продуктов. – Методы организации работы в команде разработчиков.	– Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	– Использовать выбранную систему контроля версий. – Использовать методы для получения кода с заданной функциональностью и степенью качества. – Анализировать проектную и техническую документацию. – Организовывать постобработку данных. Приемы работы в системах контроля версий. – Выявлять ошибки в системных компонентах на основе спецификаций	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного обеспечения. – Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. – Стандарты качества программной документации. – Основы организации инспектирования и верификации. – Встроенные и основные специализированные инструменты анализа качества программных продуктов. – Методы организации работы в команде разработчиков.	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

Раздел 1. Зачетно – экзаменационные материалы

Вопросы для устного опроса по темам

1 Вопросы по разделу «Основные принципы алгоритмизации и программирования»

1. Основные понятия информатики.
2. Классификация и обзор программного обеспечения ПК.
3. Алгоритм: понятие, требования, способы разработки и записи.
4. Обзор типовых алгоритмов.
5. Языки программирования. Технология программирования.
6. Понятие о модульном принципе. Текст, объектный и исполнимый модуль.

2. Вопросы по разделу «Введение в программирование и основы языка

1. Структура программы на языке: заголовок, декларативная часть, исполняемая часть.
2. Среда программирования. Состав интегрированной системы.
3. Основные понятия языка: основные символы; константы, символические имена; переменные, стандартные функции, выражения.
4. Типы данных, описание типов. Выражения, порядок вычисления выражений.
5. Классификация операторов. Операторы ввода - вывода и присваивания.
6. Обзор операторов управления. Организация ветвлений и циклов.
7. Подпрограммы. Работа с библиотеками.
8. Работа с файлами.

3. Вопросы по разделу «Основы программирования в среде»

1. Основные понятия ООП.
2. Объектно-ориентированная парадигма программирования.
3. Объекты, полиморфизм и наследование.
4. Объектно-ориентированное проектирование.
5. Библиотеки объектов.
6. Интерфейсные объекты: управляющие элементы, окна, диалоги.
7. События и сообщения.
8. Механизмы передачи и обработки сообщений в объектно-ориентированных средах.
9. Конструирование программ на основе иерархии объектов.
10. Технология разработки приложений в среде визуального программирования.
11. Знакомство со средой.
12. Конструирование объектов: строки, стеки, списки, очереди, деревья.
13. Математические объекты: рациональные и комплексные числа, вектора, матрицы.
14. Библиотеки объектов.
15. Интерфейсные объекты: управляющие элементы, окна, диалоги.
16. События и сообщения.
17. Механизмы передачи и обработки сообщений в объектно - ориентированных средах.
18. Конструирование программ на основе иерархии объектов. Разработка приложений в среде.

Критерии оценки:

оценка «отлично» выставляется студенту, если:

- полно раскрыл содержание вопроса;

- изложил материал грамотным языком в определенной логической последовательности, точно используя специализированную терминологию и символику;

- правильно выполнил графическое изображение алгоритма и иные чертежи и графики, сопутствующие ответу;

- показал умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации при выполнении практического задания;

- продемонстрировал усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость используемых при ответе умений и навыков;

- отвечал самостоятельно без наводящих вопросов преподавателя;

оценка «хорошо», если:

- ответ удовлетворяет в основном требованиям на оценку «5», но при этом имеет один из недостатков:

- в изложении допущены небольшие пробелы, не исказившие логического и информационного содержания ответа;

- допущены один-два недочета при освещении основного содержания ответа, исправленные по замечанию учителя;

- допущены ошибка или более двух недочетов при освещении второстепенных вопросов или в выкладках, легко исправленные по замечанию преподавателя;

оценка «удовлетворительно», если:

- неполно или непоследовательно раскрыто содержание вопросов билета, но показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения программного материала, имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, чертежах, блок-схем и выкладках, исправленные после нескольких наводящих вопросов преподавателя;

- студент не справился с применением теории в новой ситуации при выполнении практического задания, но выполнил задания обязательного уровня сложности по данной теме,

- при знании теоретического материала выявлена недостаточная сформированность основных умений и навыков;

оценка «неудовлетворительно», если:

- не раскрыто основное содержание программного материала;

- обнаружено незнание или непонимание студентом большей или наиболее важной части программного материала,

- допущены ошибки в определении понятий, при использовании терминологии, в чертежах, блок-схем и иных выкладках, которые не исправлены после нескольких наводящих вопросов преподавателя.

Варианты заданий для проведения проверочных работ

Тема 1.4 Методы программирования

Вариант 1

1. Записать форматированный вывод величин:

R=17.42; c=0.0001342; k=12.

2. Составить программу, которая выводила бы на экран в пять строк следующий текст:

- *Системные программы*
- *Прикладные программы*
- *Инструментальные системы или системы программирования*

3. Запишите на языке Паскаль условие, при котором треугольник со сторонами a, b, c является равносторонним.

4. Найти средний балл по дисциплине математика в группе из 20 студентов

5. Укажите типы полей в ИС *Спорт.* Минимальный список характеристик:

Фамилия спортсмена, дата рождения, вид спорта, команда, страна, зачетный результат, является ли он достижением, каким (мировой рекорд, олимпийский и т.п.) и за какой год; примечание.

Вариант 2

1. Записать выражения по правилам языка:

$$x1 = \frac{-b + \sqrt{d}}{2a}$$

$$x1 = \frac{-b - \sqrt{d}}{2a}$$

y = -2, 7· $x^3 + 0{,}23 \cdot x^2 - 1{,}4$

s=2 · $\pi \cdot r(h + r)$

2. Определить значения логических выражений:

a< >b

c > d при a=3, 14; b=2; c=6; d=4

3. Какое значение примет x при k=2 согласно фрагмента программы:

```
Case k of
        1: x:=x+5;
        2: x:=x+10;
        3: x:=x+20
else x:=0
end;
```

4. Найти средний балл по дисциплине математика в группе из 20 студентов

5. Укажите типы полей в ИС *География.* Минимальный список характеристик:

Название страны, столица, площадь территории, является ли страна развитой в экономическом отношении, количество населения, преобладающая национальность, примечание

Вариант 3

1. Перевести число в экспоненциальной форме к обычному представлению с фиксированной точкой:

-6.42e-2

-3.2e-6

2. Указать форму оператора if в каждом случае:

a) if <условие > then <оператор_1> else <оператор_2>;

b) if < условие> then < оператор>;

c) if <условие> then

```
begin
        < оператор_1>; < оператор_2>;…< оператор_n>;
end
        else
begin
        < оператор_1>; < оператор_2>; …< оператор_n>;
end;
```

3. Проверить, принадлежит ли точка с заданными координатами третьей координатной четверти. Составить блок – схему и программу.

4. Определить количество дней месяца с температурой выше средней.

5. Укажите типы полей в ИС *Персональные ЭВМ.* Минимальный список характеристик:

Фирма-изготовитель, тип процессора, тактовая частота, объем ОЗУ, объем жесткого диска, дата выпуска.

Вариант 4

1. Описать переменные: a=2,078; b=1/3; c=0,025; n=3; i=5; k=9.
2. Найти значения логических выражений:

a) a>b при a=0,5; b=0,4;

b) a+c<0 при c=1,2; d=1/4; f=0,25.

c) d< >f.

3. Составить программу, которая проверяет, является ли введенное с клавиатуры число одновременно положительным и четным.
4. В массиве 12 целых чисел заменить все отрицательные значения квадратами порядковых номеров.
5. Укажите типы полей в ИС *Библиотека.* Минимальный список характеристик:

Автор книги, название, год издания, цена, является ли новым изданием, краткая аннотация.

Тема 2.6 Организация ввода-вывода данных. Работа с файлами

1 Вариант

1. Определить на каких местах стоит буква "р" и "п".
2. Нетипизированные файлы.
3. Процедуры

2 Вариант

1. Написать программу, выдающие номер позиции заданной строки, в которой впервые встречается буква "a" и сообщение "end", если в строке нет такой буквы.
2. Средства для работы с типизированными файлами.
3. Функции

3 Вариант

1. Даны два слова. Определить, сколько начальных букв первого слова совпадают с начальными буквами второго слова.
2. Процедуры и функции обработки текстовых файлов.
3. Подпрограммы

4 Вариант

1. Дано предложение, в котором есть несколько букв «е». Найти порядковый номер первой и последней из них.
2. Запись в файл
3. Одномерные и двумерные массивы.

Тема 2.7 Библиотеки подпрограмм.

Вариант 1

1. Логические основы алгоритмизации (конъюнкция, дизъюнкция, инверсия, эквиваленция, импликация)
2. Структура программы на языке.
3. Вычислите:

$$\frac{(a-b)^3-(a^3)}{b^3-3ab^2-3a^2b},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Составить программу, которая определяет, является ли счастливым трамвайный билет (т.е. равна ли сумма первых трех цифр числа, из которых оно состоит, сумме последних трех - число шестизначное).

Вариант 2

1. Понятие алгоритма, свойства алгоритма, схемы алгоритмов. Иерархия объектов и группа. Итераторы.
2. Выражения и операции языка.
3. Вычислите:

$$\frac{(a-b)^3-(a^3-3ab^2)}{b^3-3a^2b},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник. Определить его вид (остроугольный, прямоугольный, тупоугольный).

Вариант 3

1. Языки программирования: эволюция, классификация.
2. Оператор присваивания
3. Вычислите:

$$\frac{(a-b)^3-(a^3-3a^2b)}{b^3-3ab^2},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Даны три числа a, b, c. Определить, какое из них равно некоторому числу d. Если среди данных чисел таких нет, то найти max(d-a,d-b,d-c).

Вариант 4

1. Системы программирования. Модульное программирование.
2. Составной оператор
3. Вычислите:

$$\frac{(a+b)^3-(a^3)}{3ab^2+b^3+3a^2b},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Две прямые описываются уравнениями: a1x+b1y+c1=0; a2x+b2y+c2=0. Напечатать координаты точки пересечения этих прямых, либо сообщить, что эти прямые совпадают, не пересекаются или вовсе не существуют.

Вариант 5

1. Объектно-ориентированный подход к программированию (наследование, инкапсуляция, полиморфизм, компоненты).
2. Процедуры ввода и вывода данных.
3. Вычислите:

$$\frac{(a+b)^3-(a^3+3a^2b)}{3ab^2+b^3},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Два прямоугольника заданы длинами сторон. Определите, можно ли первый прямоугольник целиком разместить во втором.

Вариант 6

1. Общие принципы разработки ПО (частотный принцип, принцип модульности, принцип функциональной избыточности, генерируемости, функциональной избыточности, «по умолчанию»).
2. Оператор условного перехода.
3. Вычислите:

$$\frac{(a-b)^2-(a^2-2ab)}{b^2},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Найти количество положительных чисел среди четырех целых чисел. Если количество положительных чисел больше, чем отрицательных, то найти наибольшее среди отрицательных, иначе наибольшее среди положительных. Посчитать количество нулевых чисел.

Вариант 7

1. Лексика языка. Переменные и константы. Типы данных.
2. Оператор безусловного перехода. Условный оператор Case.
3. Вычислите:

$$\frac{(a+b)^2-(a^2+2ab)}{b^2},$$

при a=1000, b=0.0001

4. Оператор условной передачи управления:

Даны два действительных числа, не равные друг другу. Меньшее из них заменить половиной их суммы, а большее – удвоенным произведением.

Критерии оценки:

оценка «отлично» выставляется студенту, если:

- полно раскрыл содержание вопросов варианта;
- изложил материал грамотным языком в определенной логической последовательности, точно используя специализированную терминологию и символику;
- правильно выполнил графическое изображение алгоритма и иные чертежи и графики, сопутствующие ответу;
- показал умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации при выполнении практического задания;
- продемонстрировал усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость используемых при ответе умений и навыков;
- отвечал самостоятельно без наводящих вопросов преподавателя;

оценка «хорошо», если:

- ответ удовлетворяет в основном требованиям на оценку «5», но при этом имеет один из недостатков:
- в изложении допущены небольшие пробелы, не исказившие логического и информационного содержания ответа;
- допущены один-два недочета при освещении основного содержания ответа, исправленные по замечанию учителя;
- допущены ошибка или более двух недочетов при освещении второстепенных вопросов или в выкладках, легко исправленные по замечанию преподавателя;

оценка «удовлетворительно», если:

- неполно или непоследовательно раскрыто содержание вопросов билета, но показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения программного материала, имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, чертежах, блок-схем и выкладках, исправленные после нескольких наводящих вопросов преподавателя;

- студент не справился с применением теории в новой ситуации при выполнении практического задания, но выполнил задания обязательного уровня сложности по данной теме,

- при знании теоретического материала выявлена недостаточная сформированность основных умений и навыков;

оценка «неудовлетворительно», если:

- не раскрыто основное содержание программного материала;

- обнаружено незнание или непонимание студентом большей или наиболее важной части программного материала,

- допущены ошибки в определении понятий, при использовании терминологии, в чертежах, блок-схем и иных выкладках, которые не исправлены после нескольких наводящих вопросов преподавателя.

Раздел 2. Оценочные материалы для оценки сформированности компетенций

Осваиваемая компетенция: ОК-1 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

Индикаторы достижения компетенций

Знание: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте;
алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности

Умение: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;
составить план действия; определить необходимые ресурсы;
владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)

1. **Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:**

Соотнесите задачу отбора и сортировки данных в рамках разных проектов с наиболее подходящим алгоритмом

Задача отбора и сортировки данных	Алгоритм
A: Поиск элемента в отсортированном массиве.	1. двоичный поиск
B: Поиск кратчайшего пути в графе с неотрицательными весами.	2. Dijkstra
C: Сортировка большого массива целых чисел.	3. быстрая сортировка
D: Оптимизация распределения ресурсов при заданных ограничениях (как задача типа оптимизационной DP).	4. динамическое программирование

Ответ: A1B2C3D4

Соотнесите структуру данных с подходящими операциями.

Структуры данных	Операции
1. Массив	a) быстрый доступ по индексу
2. Связанный список	b) эффективный поиск и вставка произвольного элемента
3. Дерево поиска	c) упорядоченный обход и поиск по диапазону
4. Хэш-таблица	d) среднее время доступа/поиска по ключу

Ответ: a1b2c3d4

Установите соответствие между структурами данных и их применениями:

Структуры	*Применение*
1. Стек	A) Кэширование данных
2. Очередь	B) Поиск кратчайшего пути
3. Хеш-таблица	C) Обработка вызовов функций
4. Граф	D) Буферизация задач
	E) Сортировка массивов

Ответ: 1C2D3A4B

2. **Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:**

Установите последовательность разработки программы:

1) Анализ требований
2) Проектирование алгоритма
3) Написание кода
4) Тестирование
5) Оптимизация

Ответ: 12345

Установите этапы решения задачи методом динамического программирования:

1) Определение подзадач
2) Формулировка рекуррентного соотношения
3) Создание таблицы значений
4) Заполнение таблицы
5) Восстановление решения

Ответ: 12345

3. **Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:**

Какие алгоритмы сортировки эффективны для больших данных?

a) Пузырьковая сортировка
b) Быстрая сортировка
c) Сортировка слиянием
d) Линейный поиск
e) Последовательный перебор

Ответ: bc

Какие структуры данных оптимальны для поиска информации?

a) Хеш-таблицы
b) Односвязные списки
c) Бинарные деревья поиска
d) Индексированные массивы
e) Стек

Ответ: acd

4. **Прочитайте текст и дайте ответ:**

Разработайте алгоритм для системы учета рабочего времени сотрудников с учетом различных сценариев (опоздания, переработки, больничные). Опишите структуру данных.
Пример ответа:
Алгоритм:
1. Входные данные:
Расписание (график работы)
Фактические отметки времени
Статусы (больничный/отпуск)
2. Обработка:
Сравнение с графиком
Расчет отклонений
Учет особых статусов
3. Вывод:
Отчеты по сотрудникам
Статистика по отделам
Структуры данных:
Хеш-таблица для быстрого доступа к данным сотрудников
Очередь для обработки событий
Граф для анализа зависимостей в расписании

Опишите, как можно оптимизировать алгоритм поиска по базе из 1 млн записей. Какие структуры данных и методы использовать?
Пример ответа:
Методы оптимизации:
1. Индексация по ключевым полям (B-деревья)
2. Кэширование частых запросов (хеш-таблицы)
3. Партиционирование данных
4. Использование полнотекстового поиска (Elasticsearch)
5. Оптимизация запросов:
Предварительная фильтрация
Ленивая загрузка
Пейджинг

5. Прочитайте текст и дополните его:

Алгоритм — это последовательность действий, которые приводят к _____.
Ответ: результату.
Пояснение: алгоритм задаёт шаги, которые преобразуют входные данные в ожидаемый итоговый результат.

Линейный поиск — это метод найти элемент, перебирая элементы один за другим до _____
Ответ: конца массива или конца списка.

Осваиваемая компетенция: ОК-2 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

Индикаторы достижения компетенций

Знание: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации

Умение: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска

1. **Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:**

Установите соответствие между структурами данных и их свойствами:

Структуры	Свойства
A) Массив	1. FIFO (First In - First Out)
B) Стек	2. LIFO (Last In - First Out)
C) Очередь	3. Прямой доступ по индексу
D) Дерево	4. Иерархическая организация данных
E) Переменная	

Ответ: A3B2C1D4

Установите соответствие между парадигмами программирования и их особенностями:

Парадигмы	Особенности
A) Процедурное	1. Использование объектов и классов
B) ООП	2. Последовательность вызовов процедур
C) Функциональное	3. Вычисления как оценка функций
D) Логическое	4. Основано на формальной логике
E) Графическое	

Ответ: A2B1C3D4

2. **Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:**

Установите последовательность этапов компиляции программы:

1. Лексический анализ
2. Синтаксический анализ
3. Семантический анализ
4. Генерация объектного кода
5. Линковка

Ответ: 12345

Установите последовательность выполнения программы простой программы, которая читает входные данные, обрабатывает их и выводит результат.

1. вывести результат пользователю
2. прочитать входные данные

3. обработать данные (применить алгоритм)
4. завершить работу программы

Ответ: 2314

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие парадигмы программирования существуют?

a) Объектно-ориентированное
b) Функциональное
c) Процедурное
d) Художественное
e) Музыкальное

Ответ: abc

Какие структуры данных относятся к линейным?

a) Массив
b) Дерево
c) Граф
d) Стек
e) Хэш-таблица

Ответ: ad

4. Прочитайте текст и дайте ответ:

Каковы основные различия между рекурсивными и итеративными алгоритмами? Приведите примеры, когда предпочтительнее каждый из подходов.

Ответ: Рекурсия использует вызовы функций и стек, удобна для задач с естественной рекурсивной структурой (обход деревьев). Итерация работает через циклы, эффективнее по памяти. Примеры: рекурсия - алгоритмы "разделяй и властвуй", итерация - обработка массивов.

Объясните принцип работы хеш-таблиц. Какие методы разрешения коллизий вы знаете?

Ответ: Хеш-таблица преобразует ключ в индекс массива через хеш-функцию. Методы разрешения коллизий: цепочки (списки в ячейках), открытая адресация (поиск следующей свободной ячейки), двойное хеширование.

5. Прочитайте текст и дополните его:

Процесс систематического сбора, анализа и интерпретации _____ из внешних и внутренних источников с целью поддержки решений.

Ответ: информации.

Пояснение: информация — это содержимое, которое собирают и анализируют; именно она служит основой для выводов и решений в проектной работе.

Оценка источников информации — это процесс определения доверия и достоверности _____.
Ответ: источников.

Осваиваемая компетенция: ОК-5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

Знание: особенности социального и культурного контекста; правила оформления документов и построения устных сообщений

Умение: грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе

1. **Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:**

Установите соответствие между элементами кода и комментариями:

Элементы кода	Варианты комментариев
1. Сложный алгоритм	а) Описание логики
2. Обработка ошибок	б) Возможные сценарии
3. Константы	в) Значения параметров
4. Имена переменных	

Ответ: 1а2б3в

Установите соответствие между документами и стандартами:

Документ	Варианты стандартов
1. Техническое задание	а) ГОСТ 34
2. Руководство пользователя	б) ГОСТ Р 7.0.97
3. Описание API	в) OpenAPI
4. Личные заметки	

Ответ: 1а2б3в

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность документирования кода:
1. Анализ логики программы
2. Добавление комментариев на русском языке
3. Проверка ясности пояснений
4. Согласование с командой
5. Фиксация в системе контроля версий

Ответ: 12345

Установите порядок составления ТЗ на разработку ПО:
1. Сбор требований заказчика
2. Формализация требований
3. Проверка соответствия ГОСТ 34
4. Согласование с заинтересованными сторонами

5. Утверждение документа

Ответ: 12345

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие элементы кода требуют грамотных комментариев на русском?

a) Сложные алгоритмы
b) Неочевидные решения
c) Имена переменных
d) Обработка ошибок
e) Простые циклы

Ответ: abd

Какие документы программиста должны быть на русском языке?

a) Техническое задание
b) Личные заметки
c) Инструкция для пользователей
d) Переписка в мессенджере
e) Дневник сновидений

Ответ: ac

4. Прочитайте текст и дайте ответ:

Как следует оформлять комментарии в коде для русскоязычной команды разработчиков?

Ответ: Рекомендации по оформлению:

1. Использовать профессиональную терминологию
2. Соблюдать баланс между лаконичностью и полнотой
3. Учитывать уровень подготовки команды
4. Приводить ссылки на российские стандарты
5. Использовать принятые в команде conventions

Какие особенности русского языка важно учитывать при написании технических заданий?

Ответ: Ключевые особенности:

1. Точность формулировок требований
2. Соответствие ГОСТ 34.602-89
3. Учет особенностей русской грамматики
4. Однозначность терминологии
5. Логичность структуры документа

5. Прочитайте текст и дополните его:

Деловая переписка — это форма коммуникации, направленная на решение профессиональных задач через письменную форму, с соблюдением _____, формального стиля и точности формулировок.

Ответ: этикета (или культурного этикета).

Пояснение: этикет обеспечивает корректность обращения, вежливость и структурированность текста, что критично для восприятия документов государственными и региональными организациями. Соблюдение этикета снижает риск недоразумений и повышает доверие к документу.

Устная презентация — это структурированное изложение информации с целью передачи смысла аудитории, которое требует адаптации под культурный контекст и уровень _____ аудитории.

Ответ: формальности.

Осваиваемая компетенция: ОК-9 Использовать информационные технологии в профессиональной деятельности

Индикаторы достижения компетенций

Знание: современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности

Умение: применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Установите соответствие между структурами данных и их характеристиками:

Структуры данных	**Характеристики**
1. Массив	a) LIFO (последним пришел - первым вышел)
2. Стек	b) Прямой доступ по индексу
3. Граф	c) Сетевая модель отношений
4. HTML	d) Язык разметки

Ответ: 1b2a3c

Установите соответствие между парадигмами программирования и языками:

Парадигмы	**Языки**
1. Объектно-ориентированная	a) Java
2. Функциональная	b) Haskell
3. Логическая	c) Prolog
4. Табличная	d) Photoshop

Ответ: 1a2b3c

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность этапов разработки программного обеспечения:

1. Сбор требований
2. Проектирование архитектуры
3. Написание кода
4. Тестирование

5. Внедрение

Ответ: 12345

Установите последовательность выполнения операций в выражении (по приоритету):

1. Скобки ()
2. Умножение *
3. Сложение +
4. Возведение в степень **

Ответ: 1423

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Сравните алгоритмы сортировок по требованиям к данным и памяти.

A. Вставочная сортировка эффективна для небольших массивов или когда данные почти отсортированы.

B. Быстрая сортировка неизбежно стабильно сортирует элементы по умолчанию.

C. Сортировка выбором не требует дополнительной памяти. D. Сортировка слиянием стабильно сортирует и требует дополнительной памяти.

E. Пузырьковая сортировка быстрее быстрой сортировки на больших данных.

Ответ: ADC.

Какие принципы ООП реализованы в следующем коде?

```
python
Copy
Download
class Animal:
    def __init__(self, name):
        self.__name = name

    def make_sound(self):
        pass

class Dog(Animal):
    def make_sound(self):
        return "Гав!"
```

a) Инкапсуляция
b) Наследование
c) Полиморфизм
d) Абстракция

Ответ: ab

4. Прочитайте текст и дайте ответ:

Опишите основные этапы разработки алгоритма для решения задачи автоматизации отчетности в компании. Приведите примеры инструментов/языков программирования для каждого этапа.

Ответ:

1. Анализ задачи – определение требований к отчету (Excel, интервью с заказчиком).
2. Проектирование алгоритма – блок-схемы, псевдокод (Draw.io, Lucidchart).
3. Реализация – написание кода (Python с библиотеками Pandas, OpenPyXL).
4. Тестирование – проверка на тестовых данных (pytest, ручное тестирование).
5. Внедрение – интеграция с системой (настройка CRON для автоматического запуска).

Какие принципы ООП нарушены в следующем коде? Предложите исправленную версию.

```
python
Copy
Download
class User:
    def __init__(self, name, email):
        self.name = name
        self.email = email

    def save_to_db(self):
        # код сохранения в БД
        pass

    def send_email(self, message):
        # код отправки email
        pass
```

Ответ: Нарушен **принцип единственной ответственности (SOLID)** – класс User занимается и хранением данных, и работой с БД, и отправкой email.

Исправление:

```
python
Copy
Download
class User:
    def __init__(self, name, email):
        self.name = name
        self.email = email

class UserRepository:
    def save_to_db(self, user):
        # код сохранения в БД
        pass

class EmailService:
    def send_email(self, user, message):
        # код отправки email
        pass
```

5. Прочитайте текст и дополните его:

Обработка данных — это последовательность действий по сбору, очистке и преобразованию _____ в полезные сведения.

Ответ: информации (или данных, в зависимости от формулировки).

Информационная безопасность — это совокупность мер по защите _____ информационных активов и информационных систем от несанкционированного доступа, утечки и повреждений.

Ответ: конфиденциальности (или целостности/доступности, в зависимости от акцента курса).

Осваиваемая компетенция: ОК-10 Пользоваться профессиональной документацией на государственном и иностранном языках

Индикаторы достижения компетенций

Знание: правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности

Умение: понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы

1. **Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:**

Установите соответствие между языками программирования и их типами (один лишний):

Языки	*Типы*
1. Python	а) Объектно-ориентированный
2. Java	б) Функциональный
3. SQL	в) Язык запросов
4. C++	г) Язык разметки
5. HTML	

Ответ: 1а2а3в4а5г

Соотнесите структуры данных с их характеристиками:

Структуры	*Характеристики*
1. Массив	а) LIFO (Last In, First Out)
2. Стек	б) Фиксированный размер
3. Граф	в) Ключ-значение
4. Хэш-таблица	г) Вершины и рёбра
5. Алгоритм	

Ответ: 1б2а3г4в

2. **Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:**

Установите правильную последовательность этапов разработки программы:

1. Написание кода
2. Тестирование
3. Анализ требований
4. Проектирование архитектуры
5. Документирование

Ответ: 34125

Расположите в правильном порядке этапы выполнения программы компилятором:

1. Генерация объектного кода
2. Лексический анализ
3. Оптимизация кода
4. Синтаксический анализ
5. Семантический анализ

Ответ: 24531

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из перечисленных структур данных относятся к линейным?

a) Дерево
b) Массив
c) Связный список
d) Граф
e) Стек

Ответ: bce

Какие парадигмы программирования поддерживает язык Python?

a) Объектно-ориентированное программирование
b) Функциональное программирование
c) Процедурное программирование
d) Чисто функциональное программирование
e) Аспектно-ориентированное программирование

Ответ: abc

4. Прочитайте текст и дайте ответ:

Опишите разницу между процедурным и объектно-ориентированным программированием. Приведите примеры языков для каждого подхода.

Ответ:

- Процедурное программирование основано на последовательном выполнении инструкций (функций). Примеры языков: C, Pascal.
- ООП использует объекты, инкапсуляцию, наследование и полиморфизм. Примеры языков: Java, Python, C++.

Объясните, что такое рекурсия в программировании. Приведите пример задачи, где она уместна.

Ответ: Рекурсия — вызов функцией самой себя. Пример: вычисление факториала (n! = n * (n-1)!), обход дерева.

5. Прочитайте текст и дополните его:

Профессиональная документация — это набор документов на разных языках, в котором критически важна единая _____, обеспечивающая точность передачи технических требований.

Ответ: терминология.

_____ технических терминов — это процесс адаптации словаря и выражений из одного языка в другой с сохранением точности технических значений.

Ответ: перевод.

Осваиваемая компетенция: ПК-1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

Индикаторы достижения компетенций

Знания:

Основные этапы разработки программного обеспечения.

Основные принципы технологии структурного и объектно-ориентированного программирования.

Актуальная нормативно-правовая база в области документирования алгоритмов.

Умения:

Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

Оформлять документацию на программные средства.

Оценка сложности алгоритма.

Практический опыт:

Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования

1. **Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:**

Установите соответствие между типами данных и их описанием.

Тип данных		**Описание**	
1)	int	a)	Целое число со знаком
2)	float	b)	Число с плавающей точкой одинарной точности
3)	char	c)	Символьный тип

Тип данных		Описание	
4)	bool	d)	Логический тип (истина/ложь)
5)	string	e)	Строковый тип

Ответ: 1a2b3c4d5e

Установите соответствие между этапами разработки ПО и их содержанием.

Этап		Содержание	
1)	Анализ требований	a)	Определение функционала и ограничений системы
2)	Проектирование	b)	Создание архитектуры и алгоритмов
3)	Реализация	c)	Написание кода
4)	Тестирование	d)	Проверка на соответствие требованиям
5)	Документирование	e)	Описание API и пользовательских инструкций
6)	Маркетинг	f)	Продвижение продукта на рынке

Ответ: 1a2b3c4d5e6

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность этапов разработки программы:

1. Тестирование и отладка
2. Анализ задачи и проектирование алгоритма
3. Реализация кода
4. Поддержка и сопровождение

Ответ: 2314

Расположите в правильном порядке этапы выполнения программы на языке программирования:

1. Компиляция
2. Линковка (сборка прогшраммы из объектов и библиотек)
3. Препроцессинг (этап подготовки исходного кода перед компиляцией)
4. Исполнение

Ответ: 3124

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие принципы необходимо учитывать при разработке алгоритмов согласно ТЗ?

a) Четкое соответствие требованиям технического задания

b) Оптимальность вычислительной сложности
c) Использование самых современных языков программирования
d) Корректная обработка граничных условий
e) Максимальная компактность кода

Ответ: abd

Какие инструменты помогают проектировать алгоритмы?

a) Блок-схемы и UML-диаграммы
b) Системы контроля версий
c) Псевдокод
d) Базы данных
e) Средства мониторинга

Ответ: ac

4. Прочитайте текст и дайте ответ:

Опишите основные этапы разработки программного модуля на языке Python в соответствии с техническим заданием. Какие инструменты и методы вы будете использовать на каждом этапе?

Ответ:

1. Анализ требований – изучение ТЗ, выделение входных/выходных данных, ограничений.
2. Проектирование – разработка блок-схемы или псевдокода, выбор структур данных.
3. Реализация – написание кода (функции, классы), использование IDE (PyCharm, VS Code).
4. Тестирование – модульные тесты (unittest, pytest), отладка.
5. Документирование – описание API (docstrings), руководство пользователя.

Какие принципы SOLID применяются при проектировании программного модуля? Приведите пример нарушения и исправления одного из принципов на языке Java/C++.

Ответ:

1. Принцип единственной ответственности (SRP)**:** Класс должен решать одну задачу.
2. Нарушение: Класс Order обрабатывает заказ и сохраняет его в БД.
3. Исправление: Вынести сохранение в отдельный класс OrderRepository.

5. Прочитайте текст и дополните его:

Алгоритм разработки модулей — это последовательность шагов, направленная на создание _____ модулей в рамках технического задания.»

Ответ: функциональных.

Декомпозиция задачи — это процесс разбиения общего функционала на _____ модули.

Ответ: независимые (или взаимосвязанные, если считать в контексте зависимостей).

Осваиваемая компетенция: ПК-1.2. Разрабатывать программные модули в соответствии с техническим заданием.

Индикаторы достижения компетенций

Знания:
Основные этапы разработки программного обеспечения.
Основные принципы технологии структурного и объектно-ориентированного программирования.
Знание API современных мобильных операционных систем

Умения:
Создавать программу по разработанному алгоритму как отдельный модуль.
Оформлять документацию на программные средства.
Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.

Практический опыт:
Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.
Разрабатывать мобильные приложения

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Установите соответствие между типами алгоритмов и их характеристиками. Один элемент лишний.

Алгоритм	**Характеристика**
1) Линейный	a) Последовательное выполнение операций
2) Циклический	b) Многократное повторение блока команд
3) Рекурсивный	c) Вызов алгоритмом самого себя
4) Параллельный	d) Одновременное выполнение процессов
5) Декларативный	e) Описание результата без указания способа

Ответ: 1a2b3c4d5

Соотнесите элементы ООП с их определениями. Один элемент лишний.

Элемент	**Определение**
1) Инкапсуляция	a) Сокрытие реализации
2) Полиморфизм	b) Разное поведение при одинаковом вызове
3) Композиция	c) Объединение объектов в целое
4) Итерация	d) Последовательный перебор элементов

Элемент	Определение
5) Наследование	e) Переиспользование свойств родителя

Ответ: 1a2b3c5e

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность этапов разработки программного модуля:

1. Написание кода
2. Проектирование архитектуры
3. Анализ требований ТЗ
4. Тестирование и отладка
5. Документирование

Ответ: 32145

Расположите в правильном порядке этапы обработки исключений в программе:

1. Генерация исключения
2. Обработка исключения в блоке catch
3. Выполнение блока finally
4. Объявление блока try

Ответ: 4123

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из перечисленных принципов относятся к SOLID (набор пяти принципов проектирования, направленных на создание гибкой, расширяемой и устойчивой к изменениям архитектуры объектно-ориентированной системы)**?**

a) DRY (Don't Repeat Yourself, принцип по повторному использованию кода и уменьшению дублирования)

b) Инкапсуляция

c) Принцип открытости/закрытости (OCP, существующий код должен быть открыт для расширения, но закрыт для изменения)

d) Принцип подстановки Барбары Лисков (LSP, объекты любого подкласса должны без проблем подменять объекты базового класса в любых местах программы, не нарушая корректность её поведения)

e) KISS (Keep It Simple, Stupid, принцип простоты и минимализма, который помогает писать понятный код)

Ответ: cd

Какие из следующих структур данных являются древовидными?

a) Массив

b) Связный список

c) Бинарное дерево поиска

d) Хеш-таблица

e) Граф

Ответ: ce

4. Прочитайте текст и дайте ответ:

Опишите процесс преобразования требований из технического задания в архитектуру программного модуля. Какие методики проектирования вы будете использовать и почему?

Пример ответа:

1. Анализ требований ТЗ с выделением функциональных и нефункциональных требований
2. Разработка UML-диаграмм (классов, последовательности) для визуализации структуры
3. Применение принципов SOLID для проектирования гибкой архитектуры
4. Выбор шаблонов проектирования (например, Factory для создания объектов)
5. Обоснование выбора парадигмы (ООП, функциональное программирование)

Какие подходы к тестированию программного модуля вы предложите для обеспечения соответствия ТЗ? Опишите процесс интеграционного тестирования.

Пример ответа:

- Модульное тестирование (unit tests) отдельных компонентов
- Интеграционное тестирование взаимодействия модулей
- Приемочные тесты по сценариям из ТЗ
- Использование CI/CD для автоматизации
- Пример: pytest для Python с покрытием >80%

5. Прочитайте текст и дополните его:

Разрабатывать программные модули — это создание _____ единиц, соответствующих требованиям ТЗ.»

Ответ: независимых.

Каждый модуль должен иметь чётко определённый _____, который описывает входы, выходы и поведение.»

Ответ: интерфейс (или API).

Осваиваемая компетенция: ПК-1.3. Выполнять отладку программных модулей с использованием специализированных программных средств

Индикаторы достижения компетенций

Знания: Основные принципы отладки и тестирования программных продуктов.
Инструментарий отладки программных продуктов

Умения:
Выполнять отладку и тестирование программы на уровне модуля.
Оформлять документацию на программные средства.
Применять инструментальные средства отладки программного обеспечения

Практический опыт:
Использовать инструментальные средства на этапе отладки программного продукта.
Проводить тестирование программного модуля по определенному сценарию

1. **Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:**

Установите соответствие между видами логических ошибок и инструментами для их обнаружения.

Вид ошибки	**Инструмент обнаружения**
1) Утечка памяти	a) Valgrind
2) Ошибка сегментации	b) GDB
3) Бесконечный цикл	c) Профилировщик (Profiler)
4) Синтаксическая ошибка	d) Компилятор
5) Логическая ошибка	e) Статический анализатор

Ответ: 1a2b3c4d

Соотнесите действия при отладке с соответствующими этапами процесса.

Действие	**Этап отладки**
1) Установка точек останова	a) Подготовка к отладке
2) Просмотр call stack	b) Анализ состояния программы
3) Изменение требований	c) Внесение изменений
4) Пошаговое выполнение	d) Поиск ошибки
5) Логирование значений	e) Мониторинг состояния

Ответ: 1a2b4d5e

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность действий при отладке программы:

1. Анализ стека вызовов
2. Запуск программы в режиме отладки
3. Постановка точек останова (breakpoints)
4. Пошаговое выполнение кода (step-by-step execution)
5. Анализ значений переменных

Ответ: 32451

Расположите в правильном порядке этапы обработки исключения при отладке:

1. Выполнение блока finally
2. Генерация исключения

3. Поиск подходящего обработчика исключения
4. Завершение работы программы (если исключение не обработано)
5. Выполнение блока catch

Ответ: 23514

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из перечисленных инструментов являются специализированными средствами отладки?

a) Git, система контроля версий.

b) GDB (GNU Debugger, специализированный отладчик для программ на языках C/C++)

c) Visual Studio Debugger, встроенный в IDE отладчик для проектов на .NET/C++, предоставляющий пошаговую отладку, инспекцию переменных, стек вызовов, точки останова

d) Jira, система управления задачами и проектами.

e) Valgrind, набор инструментов для динамического анализа: память, утечки, обращение к освобожденной памяти, профилирование и т.п.

f) Docker, платформа для контейнеризации приложений.

Ответ: все

Какие типы ошибок можно выявить в процессе отладки?

a) Ошибки сегментации (Segmentation Fault)
b) Логические ошибки в алгоритме
c) Ошибки в техническом задании
d) Ошибки работы с памятью (Memory Leaks)
e) Орфографические ошибки в комментариях
f) Ошибки сетевого подключения

Ответ: abd

4. Прочитайте текст и дайте ответ:

Опишите ваш алгоритм действий при обнаружении периодически возникающей ошибки в работе модуля, которую не удается воспроизвести при пошаговой отладке. Какие специализированные средства вы будете использовать и почему?

Пример ответа:

1. Включение детального логирования (логгирование состояния системы перед ошибкой)
2. Использование conditional breakpoints в отладчике для остановки при специфических условиях
3. Применение инструментов трассировки (strace/ltrace для Linux, ETW для Windows)
4. Использование средств анализа дампов памяти при падении (core dumps)
5. Написание модульных тестов для изоляции проблемы
6. Применение реверс-отладки (например, в GDB или WinDbg)

Какие методы и инструменты вы предложите для ускорения отладки сложного многопоточного приложения? Опишите стратегию поиска гонки условий (race condition).

Пример ответа:

1. Инструменты:
ThreadSanitizer (TSan) для обнаружения гонок данных
Helgrind из набора Valgrind
Визуализаторы выполнения потоков в IDE
2. Стратегия:
Логирование порядка блокировок мьютексов
Анализ временных меток в логах
Создание минимального воспроизводящего теста
Использование детекторов deadlock
Применение формальных методов верификации для критических секций
3. Профилактика:
Статический анализ кода на потенциальные race conditions
Использование атомарных операций и lock-free структур
Документирование потокобезопасности API

5. Прочитайте текст и дополните его:

Отладка — это процесс выполнения программы в пошаговом режиме с целью выявления и анализа _____.

Ответ: ошибок (дефектов, багов).

Логирование — это систематическая запись событий во времени, предназначенная для последующей _____ ошибок и анализа поведения программы.»

Ответ: диагностики (анализа).

Осваиваемая компетенция: ПК 1.4. Выполнять тестирование программных модулей

Индикаторы достижения компетенций

Знания:
Основные виды и принципы тестирования программных продуктов

Умения:
Выполнять отладку и тестирование программы на уровне модуля.
Оформлять документацию на программные средства

Практический опыт:
Проводить тестирование программного модуля по определенному сценарию.
Использовать инструментальные средства на этапе тестирования программного продукта

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Установите соответствие между видами тестирования и их описаниями. Один элемент лишний.

Виды тестирования	**Описания**
1. Модульное тестирование	А) Проверка взаимодействия между несколькими модулями
2. Интеграционное тестирование	В) Проверка отдельных функций или методов программы

3. Регрессионное тестирование	C) Проверка системы на устойчивость к высокой нагрузке
4. Компиляция кода	D) Проверка ранее исправленных ошибок после внесения изменений
5. Нагрузочное тестирование	

Ответ: 1B2A3D5C

Установите соответствие между этапами тестирования и используемыми инструментами. Один элемент лишний.

Этапы тестирования	**Инструменты**
1. Написание тестов	A) JUnit, pytest, это средства тестирования (unit-тесты) для разных языков: JUnit — для Java, pytest — для Python
2. Запуск тестов	B) JaCoCo, Coverage.py, это инструменты для расчета покрытия тестами (code coverage, JaCoCo — для Java, Coverage.py — для Python
3. Анализ покрытия кода	C) Git, Subversion, это системы управления версиями
4. Отладка кода	D) Debugger (GDB, pdb), это средства отладки
5. Написание документации	

Ответ: 1A2A3B4D

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность этапов модульного тестирования:

1. Написание тестовых случаев
2. Запуск тестов и анализ результатов
3. Рефакторинг кода (при необходимости)
4. Идентификация тестируемого модуля
5. Исправление ошибок

Ответ: 41253

Расположите в правильном порядке шаги разработки через тестирование (TDD):

1. Рефакторинг кода
2. Написание минимального кода для прохождения теста
3. Написание падающего теста
4. Запуск теста и проверка, что он не проходит
5. Повторение цикла для новой функциональности

Ответ: 34215

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из перечисленных видов тестирования относятся к уровню модульного тестирования?

a) Тестирование отдельных функций/методов
b) Тестирование взаимодействия между модулями

c) Проверка граничных значений
d) Тестирование пользовательского интерфейса
e) Тестирование обработки ошибок

Ответы: ace

Какие инструменты используются для автоматизации тестирования программных модулей?

a) JUnit (фреймворк для модульного тестирования Java-кода)
b) pytest (фреймворк для модульного и функционального тестирования Python-кода)
c) Google Test (фреймворк для модульного тестирования C++
d) Selenium (инструмент автоматизации тестирования веб-интерфейсов)
e) Postman (платформа и инструмент для тестирования REST/GraphQL API)

Ответ: abc

4. Прочитайте текст и дайте ответ:

Опишите процесс создания модульного теста для функции, которая проверяет, является ли число простым. Какие тестовые случаи следует предусмотреть и почему?

Ответ: Для тестирования функции isPrime(n) следует:

1. Создать тесты для граничных случаев:

n = 0 и 1 (не простые)
n = 2 (наименьшее простое)
большое простое число (например, 997)

2. Проверить четные числа > 2 (не простые)
3. Проверить нечетные составные числа (9, 15)
4. Проверить отрицательные числа (можно считать не простыми или обрабатывать ошибку)
5. Добавить тест на обработку нечислового ввода (если требуется)

Обоснование: такой набор покрывает основные логические ветви функции и пограничные ситуации.

Какие принципы следует учитывать при организации модульного тестирования в большом проекте? Как обеспечить поддержку и актуальность тестов при изменениях кода?

Ответ: Ключевые принципы:

1. Независимость тестов - каждый тест должен работать изолированно
2. Читаемость - понятные имена тестов и сообщения об ошибках
3. Полнота покрытия - тестирование всех значимых сценариев
4. Быстрота выполнения - тесты должны работать быстро
5. Поддержка актуальности:

Интеграция в CI/CD
Обязательное обновление тестов при изменении требований
Регулярный ревью тестов
Использование инструментов покрытия кода
Автоматизированный запуск перед коммитами

Дополнительно важно документировать тестовые случаи и применять практику тестирования наравне с разработкой.

5. Прочитайте текст и дополните его:

Тестирование модулей — это процесс исполнения кода с целью выявления _____ и подтверждения корректности поведения модуля.

Ответ: ошибок (дефектов, дефектов в коде).

Регрессионное тестирование — это повторное выполнение тестов, чтобы убедиться, что исправления не повлияли на существующую _____ функциональность.

Ответ: корректную (или ранее работавшую).

Осваиваемая компетенция: ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода

Индикаторы достижения компетенций

Знания:
Способы оптимизации и приемы рефакторинга.
Инструментальные средства анализа алгоритма.
Методы организации рефакторинга и оптимизации кода.
Принципы работы с системой контроля версий

Умения:
Выполнять оптимизацию и рефакторинг программного кода.
Работать с системой контроля версий

Практический опыт:
Анализировать алгоритмы, в том числе с применением инструментальных средств.
Осуществлять рефакторинг и оптимизацию программного кода

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Установите соответствие между "запахами кода" (code smells) и методами их устранения.

Запахи кода	**Методы рефакторинга**
1. Длинный метод	A) Выделение метода (Extract Method)
2. Магические числа	B) Введение параметра
3. Дублирование кода	C) Замена числа именованной константой
4. Избыточные временные переменные	D) Удаление промежуточных переменных
	E) Увеличение количества глобальных переменных

Ответ: 1A2C3A4D

Установите соответствие между видами оптимизации и примерами их применения.

Виды оптимизации	**Примеры**
1. Алгоритмическая оптимизация	A) Замена пузырьковой сортировки на быструю
2. Оптимизация памяти	B) Уменьшение размера структур данных

3. Оптимизация ввода-вывода	C) Добавление кэширования запросов
4. Преждевременная оптимизация	D) Оптимизация всех участков кода перед профилированием
	E) Использование буферизации при работе с файлами

Ответ: 1A2B3E4D

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность действий при проведении рефакторинга кода

1. Написание модульных тестов для существующего кода
2. Выявление "запахов кода" (code smells)
3. Выполнение изменений по улучшению кода
4. Проверка, что все тесты проходят после изменений
5. Анализ метрик качества кода (сложность, связность)
6. Фиксация изменений в системе контроля версий

Ответ: 251346

Установите правильный порядок шагов оптимизации производительности кода

1. Замер производительности после изменений
2. Выявление узких мест через профилирование
3. Реализация оптимизационных изменений
4. Постановка конкретных целей оптимизации
5. Документирование проведенных оптимизаций
6. Анализ необходимости дальнейших улучшений

Ответ: 423165

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из следующих действий относятся к правильным практикам рефакторинга кода?

a) Удаление дублирующегося кода
b) Увеличение длины методов для уменьшения их количества
c) Замена числовых констант именованными значениями
d) Добавление новых функций в процессе рефакторинга
e) Упрощение сложных условных выражений
f) Увеличение количества глобальных переменных для удобства

Ответ: ace

Какие из перечисленных методов можно использовать для оптимизации производительности кода?

a) Замена линейного поиска на бинарный для больших массивов
b) Кэширование результатов ресурсоемких вычислений
c) Увеличение количества временных переменных
d) Использование более сложных алгоритмов без предварительного анализа
e) Уменьшение количества вызовов к базе данных
f) Замена всех циклов на рекурсивные вызовы

Ответ: abe

4. Прочитайте текст и дайте ответ:

Опишите процесс рефакторинга сложного метода, содержащего множество вложенных условных операторов. Какие конкретные шаги вы предпримете для улучшения его читаемости и поддерживаемости? Какие паттерны или принципы программирования при этом примените?

Пример ответа:

1. Сначала проведу анализ метода, выделю отдельные логические блоки
2. Применю технику "Выделение метода" (Extract Method) для каждой самостоятельной логической части
3. Заменю вложенные условия на guard clauses (ранние возвраты) там, где это уместно
4. Рассмотрю возможность применения паттерна "Стратегия" или "Состояние" для сложной условной логики
5. Введу понятные именованные булевы переменные/методы вместо сложных условий
6. Добавлю комментарии, поясняющие бизнес-логику
7. Напишу юнит-тесты для рефакторенного кода
8. Использую принципы SOLID, особенно Single Responsibility

Вам нужно оптимизировать код, работающий с большими объемами данных. Опишите ваш подход к поиску узких мест и конкретные техники оптимизации, которые вы будете применять. Как вы убедитесь, что оптимизация дала результат и не внесла регрессий?

Пример ответа:

1. Инструментарий:

Использую профайлер (например, YourKit или VisualVM)
Замерю время выполнения до оптимизации
Выявлю "горячие точки" (hot spots)

2. Методы оптимизации:

Алгоритмическая: заменю $O(n^2)$ на $O(n \log n)$ где возможно
Кэширование повторяющихся вычислений
Оптимизация работы с памятью (пулинг объектов)
Векторизация операций
Параллелизация с помощью Stream API/CompletableFuture

3. Проверка:

Замер производительности после изменений
Сравнение метрик до/после
Проверка всех тестов (юнит, интеграционные)
Валидация результатов на реальных данных
Мониторинг в production после выкатки

4. Документирование:

Фиксация benchmark-результатов
Комментарии в коде о причинах оптимизации
Отметки в истории версий

5. Прочитайте текст и дополните его:

Рефакторинг — это процесс изменения _____ кода с целью улучшить читаемость, сопровождаемость и локализацию дефектов.

Ответ: структуры (или архитектуры, дизайна).

Оптимизация кода — это процесс улучшения _____ программы для достижения меньшего времени выполнения и меньшего использования памяти при сохранении корректности.

Ответ: эффективности (или производительности).

Осваиваемая компетенция: ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения

Индикаторы достижения компетенций

Знания:
Модели процесса разработки программного обеспечения.
Основные принципы процесса разработки программного обеспечения.
Основные подходы к интегрированию программных модулей.
Основы верификации и аттестации программного обеспечения.
Методы и способы идентификации сбоев и ошибок при интеграции приложений.
Методы и схемы обработки исключительных ситуаций.
Основные методы и виды тестирования программных продуктов.
Приемы работы с инструментальными средствами тестирования и отладки.
Стандарты качества программной документации.
Основы организации инспектирования и верификации.
Встроенные и основные специализированные инструменты анализа качества программных продуктов.
Методы организации работы в команде разработчиков

Умения:
Использовать выбранную систему контроля версий.
Анализировать проектную и техническую документацию.
Выполнять тестирование интеграции.
Организовывать постобработку данных.
Использовать приемы работы в системах контроля версий.
Оценивать размер минимального набора тестов.
Разрабатывать тестовые пакеты и тестовые сценарии.
Выполнять ручное и автоматизированное тестирование программного модуля.
Выявлять ошибки в системных компонентах на основе спецификаций

Практический опыт:
Разрабатывать тестовые наборы (пакеты) для программного модуля.
Разрабатывать тестовые сценарии программного средства.
Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Установите соответствие между видами тестов и их характеристиками.

Виды тестов	Характеристики
1. Модульные тесты	A) Проверяют взаимодействие компонентов
2. Интеграционные тесты	B) Оценивают производительность системы
3. Регрессионные тесты	C) Тестируют отдельные функции/методы
4. Нагрузочные тесты	D) Проверяют корректность интерфейса
5. UI-тесты	E) Анализируют требования к безопасности

Ответ: 1C2A34B5D

Установите соответствие между техниками тест-дизайна и примерами их применения.

Техники	Примеры применения
1. Эквивалентное разбиение	A) Проверка полей ввода на минимально/максимально допустимые значения
2. Анализ граничных значений	B) Тестирование работы лифта между этажами
3. Таблица принятия решений	C) Проверка всех возможных комбинаций условий в бизнес-правиле
4. Тестирование состояний	D) Оптимизация скорости работы алгоритма
	E) Тестирование только типовых значений из каждого класса эквивалентности

Ответ: 1E2A3C4B

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность разработки тестовых сценариев для нового функционала

1. Анализ требований к функционалу
2. Реализация тестового набора
3. Определение критериев тестирования
4. Разработка тестовых данных
5. Планирование тестового покрытия
6. Валидация тестовых сценариев

Ответ: 135426

Установите правильный порядок выполнения модульного тестирования

1. Анализ результатов тестирования
2. Написание тестового кода
3. Запуск тестов
4. Рефакторинг тестов (при необходимости)
5. Определение тестируемого модуля
6. Фиксация дефектов

Ответ: 523164

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из перечисленных характеристик должны учитываться при разработке

тестовых сценариев?

a) Возраст разработчика
b) Граничные значения входных данных
c) Цвет интерфейса приложения
d) Альтернативные пути выполнения
e) Обработка ошибочных ситуаций
f) Размер команды разработчиков

Ответ: bde

Какие из следующих утверждений о тестовом покрытии являются верными?

a) 100% покрытие кода гарантирует отсутствие ошибок
b) Покрытие должно включать как happy path, так и исключительные ситуации
c) Тестовое покрытие нужно измерять только для UI-слоя
d) Критерии покрытия должны определяться на основе критичности функционала
e) Достаточно тестировать только новые функции приложения

Ответ: bd

4. Прочитайте текст и дайте ответ:

Опишите процесс разработки тестовых сценариев для функции валидации пароля, которая должна проверять: длину (8-20 символов), наличие цифр, спецсимволов и букв разного регистра. Какие тестовые случаи вы включите и как организуете тестовые данные?

Пример ответа:

1. **Тестовые случаи для граничных значений длины:**

7 символов (недопустимо)
8 символов (минимум)
20 символов (максимум)
21 символ (недопустимо)

2. **Проверка состава символов:**

Только буквы нижнего регистра
Буквы нижнего+верхнего регистра
Буквы+цифры
Буквы+спецсимволы
Полный набор требований
Пустая строка

3. **Организация тестовых данных:**

Использование параметризованных тестов
Отдельные методы для разных аспектов проверки
Читаемые названия тестовых методов
Таблицы принятия решений для комбинаций условий

4. **Дополнительные проверки:**

Производительность при длинных паролях
Корректность сообщений об ошибках
Поведение при вводе пробелов

Как бы вы оптимизировали большой набор регрессионных тестов для ежедневного прогона? Какие критерии приоритезации будете использовать и какие инструменты применять для управления тестами?

Пример ответа:

1. **Критерии приоритезации:**

Критичность функционала для бизнеса

Частота использования фичи
История дефектов в компоненте
Сложность реализации
Зависимости между модулями

2. **Стратегии оптимизации:**

Разделение на smoke/sanity/полные тесты
Параллельный запуск независимых тестов
Приоритизация быстрых тестов
Кэширование подготовленных данных
Инкрементальное тестирование измененных модулей

3. **Инструменты:**

TestNG/JUnit для управления наборами
Jenkins/GitLab CI для планирования
Системы анализа покрытия (JaCoCo)
Хранилища тестовых данных
Средства визуализации результатов

4. **Метрики для оценки:**

Время выполнения набора
Процент обнаруженных дефектов
Стабильность тестов
Covergage-метрики
Соотношение manual/autotest

5. Прочитайте текст и дополните его:

Тестовый набор — это группа связанных тестов, направленных на проверку _____ и устойчивости программного продукта.

Ответ: полноты (или полноты охвата).

Тестовый сценарий описывает последовательность действий пользователя и _____ системы для проверки ожидаемого поведения.»

Ответ: ожидания (или критерии результата).

Осваиваемая компетенция: ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

Индикаторы достижения компетенций

Знания:

Модели процесса разработки программного обеспечения.
Основные принципы процесса разработки программного обеспечения.
Основные подходы к интегрированию программных модулей.
Основы верификации и аттестации программного обеспечения.
Стандарты качества программной документации.
Основы организации инспектирования и верификации.
Встроенные и основные специализированные инструменты анализа качества программных продуктов.
Методы организации работы в команде разработчиков

Умения:

Использовать выбранную систему контроля версий.

Использовать методы для получения кода с заданной функциональностью и степенью качества.
Анализировать проектную и техническую документацию.
Организовывать постобработку данных.
Приемы работы в системах контроля версий.
Выявлять ошибки в системных компонентах на основе спецификаций

Практический опыт:
Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Установите соответствие между типами нарушений стандартов кодирования и методами их выявления.

Типы нарушений	**Методы выявления**
1. Стилевые нарушения (отступы, форматирование)	A) Ручной review кода
2. Архитектурные проблемы (высокая связность)	B) Статический анализ (SonarQube)
3. "Запахи кода" (дублирование, длинные методы)	C) Нагрузочное тестирование
4. Нарушения безопасности (уязвимости)	D) Проверка стиля (Checkstyle)
	E) Анализ зависимостей (JDepend)

Ответ: 1D2E3B4A

Установите соответствие между инструментами анализа кода и их специализацией.

Инструменты	**Специализация**
1. Checkstyle	A) Проверка стиля Java-кода
2. PMD	B) Поиск шаблонов ошибок в коде
3. SpotBugs	C) Обнаружение потенциальных багов
4. JUnit	D) Проверка стиля JavaScript-кода
5. ESLint	E) Измерение покрытия тестами

Ответ: 1A2B3C5D

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите правильную последовательность проведения инспектирования кода
1. Автоматизированная проверка с помощью инструментов статического анализа
2. Подготовка отчета о найденных несоответствиях
3. Определение стандартов кодирования для проверки
4. Ручной review критических участков кода
5. Исправление выявленных нарушений
6. Планирование периодичности проверок

Ответ: 631425

Установите правильный порядок работы с нарушениями стандартов кодирования

1. Классификация найденных нарушений по критичности
2. Автоматическое исправление с помощью инструментов форматирования
3. Обсуждение спорных моментов с командой
4. Фиксация нарушений в системе отслеживания задач
5. Проверка исправлений
6. Назначение сроков исправления

Ответ: 134625

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие из перечисленных аспектов должны проверяться при инспектировании кода на соответствие стандартам?

a) Соблюдение naming conventions (правил именования)
b) Скорость выполнения кода
c) Правильность использования отступов и форматирования
d) Количество строк в файле
e) Наличие комментариев для сложной логики
f) Цветовая схема IDE разработчика

Ответ: ace

Какие инструменты можно использовать для автоматизации проверки кода?

a) JIRA, система трекинга задач и управления проектами. Используется для планирования и отслеживания дефектов

b) SonarQube, платформа для статического анализа кода и измерения технического долга. Проверяет качество кода по многочисленным правилам и метрикам

c) Checkstyle, инструмент статического анализа кода для Java. Проверяет стиль кода согласно заданным правилам форматирования и оформления, обнаруживает несоответствия стиля, потенциальные ошибки и т.д

d) Postman, инструмент для тестирования API

e) Photoshop, графический редактор.

f) ESLint, линтер для JavaScript/TypeScript. Анализирует исходный код на наличие ошибок, стилистических нарушений и проблем с качеством кода; поддерживает пользовательские правила и плагины, хорошо интегрируется в CI/CD и поток разработки

Ответ: bcf

4. Прочитайте текст и дайте ответ:

Опишите ваш подход к организации инспектирования кода в командном проекте. Какие этапы включает процесс? Какие инструменты автоматизированной проверки вы будете использовать? Как обеспечить, чтобы инспектирование было эффективным, но не замедляло процесс разработки?

Пример ответа:

1. **Этапы процесса:**

Предварительная настройка инструментов статического анализа

Автоматизированная проверка при каждом коммите (pre-commit hooks)

Регулярные (еженедельные) сессии ручного ревью

Классификация найденных проблем по критичности

Обязательный ревью перед мерджем в основную ветку

2. **Инструменты:**

Для Java: Checkstyle + PMD + SpotBugs

Для JavaScript: ESLint + Prettier

Для универсального анализа: SonarQube

Интеграция в CI/CD (Jenkins/GitLab CI)

3. **Оптимизация процесса:**

Автоматизация рутинных проверок

Четкие критерии для обязательного и рекомендательного исправления

Лимит времени на ревью (не более 400 строк за раз)

Использование шаблонов комментариев

Rotating lead reviewer среди команды

Как вы будете действовать при обнаружении серьезного несоответствия стандартам в legacy-коде? Какие факторы учтете при принятии решения о необходимости исправления? Как организовать процесс исправления, чтобы не нарушить работу системы?

Пример ответа:

1. **Анализ ситуации:**

Определение критичности нарушения (безопасность/стабильность/читаемость)

Оценка связанности с другими компонентами

Анализ истории изменений проблемного участка

Проверка покрытия тестами

2. **Принятие решения:**

Обязательное исправление для проблем безопасности

Постепенное исправление с рефакторингом для сложных случаев

Временное исключение для особо критичного legacy-кода

3. **Стратегия исправления:**

Создание технического долга с четким описанием

Постепенный рефакторинг через feature toggles

Обязательное покрытие тестами перед изменениями

Разбиение на небольшие atomic changes

Координация с QA командой

4. **Документирование:**

Фиксация решения в системе управления задачами

Комментарии в коде с пояснением временных решений

Обновление стандартов кодирования

5. Прочитайте текст и дополните его:

Стандарт кодирования — это набор правил и рекомендаций, обеспечивающий единый стиль _____ кода и упрощающий чтение и сопровождение.

Ответ: оформления (или форматирования).

Инспекция кода — это процесс анализа исходного кода на предмет соблюдения _____, включая именование, комментарии и модульную структуру.

Ответ: требований (или норм кодирования/правил стиля).