



«Утверждаю»

Председатель УМС

Абдрешов Ш.А.

2025г.

## ПЕРЕЧЕНЬ

вопросов и практических задач,  
выносимых на аттестационный (комплексный) экзамен  
по образовательной программе  
«6B06118 Программная инженерия»

### 1. Дисциплина «Инструментальные средства разработки программ»

1. Интегрированные среды разработки (IDE): назначение, функции и преимущества.
2. Архитектура и особенности использования Visual Studio, PyCharm и VS Code.
3. Системы контроля версий: принципы работы, основные команды Git.
4. GitHub, GitLab и Bitbucket: сравнение возможностей и сценарии применения.
5. Процессы ветвления в Git (branching models): Git Flow, Feature Branches.
6. Инструменты автоматизации сборки: Maven, Gradle, Make.
7. Система управления проектами Apache Maven: структура проекта и POM-файл.
8. Пакетные менеджеры (npm, pip, NuGet): назначение и особенности работы.
9. Управление зависимостями в программных проектах: принципы и инструменты.
10. Инструменты для документирования кода (Sphinx, Javadoc, Swagger).
11. Средства тестирования ПО: JUnit, PyTest, NUnit.
12. Типы тестирования и автоматизация тестирования.
13. Отладка программ: инструменты, методы, точки останова, просмотр переменных.
14. Логирование и мониторинг приложений: инструменты и лучшие практики.
15. Docker: основы контейнеризации и создание Dockerfile.
16. Docker Compose: управление многоконтейнерными приложениями.
17. Контейнеризация vs виртуализация: сравнение и области применения.
18. Системы непрерывной интеграции и доставки (CI/CD): принципы и пайплайны.
19. GitHub Actions, GitLab CI, Jenkins: сравнительная характеристика.
20. Управление версиями ПО и выпуск релизов.
21. Инструменты анализа качества кода: SonarQube, ESLint, Pylint.
22. UML-моделирование: инструменты (Visual Paradigm, StarUML, Enterprise Architect).
23. CASE-средства и их роль в разработке программных систем.
24. Figma и инструменты UI/UX-прототипирования в разработке ПО.
25. Средства для совместной работы в команде: Jira, Trello, Asana.
26. DevOps-подход: основные инструменты и этапы жизненного цикла.
27. Средства управления конфигурациями: Ansible, Puppet, Chef.
28. Виртуальные среды разработки: venv, Conda, VirtualBox.
29. Cloud-инструменты разработки: AWS, Azure, Google Cloud.
30. Инструменты контроля безопасности кода и статический анализ (SAST/DAST).

### 2. Дисциплина «Основы компьютерного моделирования»

1. Основные особенности языка Python: интерпретация, динамическая типизация, области применения.
2. Структура программы на Python. Переменные, типы данных и операции.



3. Управляющие конструкции: условия (if/elif/else) и циклы (for, while).
4. Строки и операции со строками. Методы строк.
5. Списки в Python: создание, индексация, срезы, основные методы.
6. Кортежи и их особенности. Отличия кортежей от списков.
7. Словари и множества: структура данных, методы, области применения.
8. Функции: объявление, параметры, области видимости, рекурсия.
9. Анонимные функции (lambda), функции высшего порядка (map, filter, reduce).
10. Работа с файлами: чтение, запись, режимы работы файлов.
11. Исключения: обработка ошибок, блоки try/except/finally.
12. Модули и пакеты: импорт, организация структуры проекта.
13. Виртуальные среды Python (venv, conda).
14. Объектно-ориентированное программирование: классы, объекты, методы.
15. Наследование, инкапсуляция, полиморфизм.
16. Магические методы (dunder methods) и перегрузка операторов.
17. Генераторы и итераторы.
18. Декораторы: назначение, принципы работы, примеры.
19. Работа с датами, временем и библиотекой datetime.
20. Работа с базами данных в Python (sqlite3, SQLAlchemy).
21. Основы обработки данных: библиотека NumPy.
22. Табличные данные и анализ данных в библиотеке pandas.
23. Визуализация данных: matplotlib, seaborn.
24. Основы сетевого программирования: sockets, HTTP-запросы (requests).
25. Основы многопоточности и многопроцессности (threading, multiprocessing).
26. Работа с API, JSON и XML.
27. Основы веб-разработки на Python: Flask/Django (структура, маршруты).
28. Тестирование Python-кода: unittest, pytest.
29. Логирование и отладка программ (logging, pdb).
30. Практики написания качественного Python-кода: PEP 8, типизация (typing).

### **3. Дисциплина «Объектно-ориентированное программирование»**

1. Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм, абстракция.
2. Понятие класса и объекта. Структура класса, свойства (атрибуты) и методы.
3. Конструкторы и деструкторы. Жизненный цикл объекта.
4. Модификаторы доступа: public, private, protected (в языках C++/Java).
5. Инкапсуляция: определение, преимущества, практические примеры.
6. Наследование: виды наследования, иерархии классов, переопределение методов.
7. Полиморфизм: статический и динамический. Примеры реализации.
8. Абстрактные классы и интерфейсы. Применение в проектировании систем.
9. Переопределение (override) и перегрузка (overload) методов.
10. Диаграммы классов UML: элементы, связи, использование в проектировании.
11. Композиция и агрегация: различия и примеры.
12. Множественное наследование: преимущества, проблемы и способы решения.
13. Исключения в ООП: обработка ошибок, иерархии исключений.
14. Статические члены класса: статические методы и переменные.
15. Виртуальные функции (в C++), механизмы динамического связывания.
16. Генерация объектов и управление памятью (new/delete, сборщик мусора).
17. Шаблоны проектирования (Design Patterns): роль и классификация.



18. Порождающие паттерны: Singleton, Factory, Abstract Factory.
19. Структурные паттерны: Adapter, Decorator, Composite.
20. Поведенческие паттерны: Strategy, Observer, Command.
21. Принципы SOLID: описание и применение в ООП.
22. Работа с коллекциями объектов: списки, словари, наборы (в разных языках).
23. Использование generics (обобщений) в Java/C# и template в C++.
24. Перегрузка операторов: назначение и примеры.
25. Интерфейсы и их отличие от абстрактных классов.
26. Механизмы сериализации объектов.
27. Обработка событий и подписчиков (event-driven programming).
28. Система модулей и пространств имён (namespace, package).
29. Принципы рефакторинга объектно-ориентированного кода.
30. Применение ООП в современных языках: Java, C#, Python, C++, Swift.

#### 4. Задачи

1. Анализ датасета о продажах (CSV + Pandas + графики)

Загрузить CSV-файл (любой: продажи, студенты, транзакции).

Выполнить:

- вывод первых строк;
- типы данных;
- пропущенные значения;
- топ-5 самых продаваемых категорий;
- построить график продаж по месяцам (line plot).

2. Статистический анализ массива данных (NumPy + визуализация)

Сгенерировать случайный массив 1000 чисел.

Вычислить:

- среднее, медиану, дисперсию, стандартное отклонение;
- построить гистограмму распределения;
- найти выбросы (значения  $> 3\sigma$ ).

3. Анализ текстов (NLP basics + Counter)

Загрузить текстовый файл (или взять текст из интернета).

Выполнить:

- очистку текста от пунктуации;
- подсчёт слов;
- частотный анализ слов (Counter);
- визуализацию топ-20 слов (bar chart).

4. Работа с API GitHub (requests)

Получить данные о пользователе GitHub:

- login, followers, public repos;
- даты создания;
- список названий репозиторий;
- вывести топ-5 репозиторий по количеству звезд.

5. Анализ данных о фильмах (pandas + группировка)

Использовать готовый датасет IMDB или TMDb (есть в публичном доступе).

Выполнить:

- топ-10 фильмов по рейтингу;
- средний рейтинг по жанрам;

- визуализация: рейтинг vs бюджет;
- найти фильмы с окупаемостью  $> 200\%$ .

#### 6. Визуализация 3D-данных (Matplotlib 3D)

Сгенерировать 3D-точки:

- $x, y$  — случайные;
- $z$  — функция  $z = \sin(x) + \cos(y)$ ;
- Построить 3D-график поверхности или scatter plot.

#### 7. Модель линейной регрессии (sklearn)

Сгенерировать данные:

- $x$  — случайные значения;
- $y = 3x + 7 + \text{noise}$ .

Выполнить:

- обучение модели LinearRegression;
- визуализацию линии регрессии;
- MAE, MSE, R2.

#### 8. Работа с JSON-файлом высокого уровня вложенности

Загрузить JSON (например, данные погоды, GitHub API или свой).

Задачи:

- вывести все ключи 1 уровня;
- найти все элементы типа "список";
- извлечь конкретные поля (температура, дата, локация);
- преобразовать в DataFrame.

#### 9. Анализ временных рядов (pandas + resample)

Использовать датасет с датами (например, цены криптовалют, обменный курс).

Сделать:

- преобразование индекса в datetime;
- агрегацию по дням/месяцам;
- скользящее среднее;
- график тренда.

#### 10. Имитационное моделирование (Monte Carlo Simulation)

Провести 10 000 симуляций:

- смоделировать цены акции по модели случайного блуждания;
- построить распределение финальных значений;
- визуализировать 50 случайных траекторий;
- найти вероятность падения ниже заданного уровня.

#### 11. Создание класса и объекта

- Создайте класс Student с полями *имя*, *группа*, *средний балл*. Реализуйте метод, который выводит полную информацию о студенте.

#### 12. Инкапсуляция и геттеры/сеттеры

Разработайте класс BankAccount с приватным полем *balance*. Реализуйте методы *deposit()* и *withdraw()* с проверкой корректности суммы.

#### 13. Использование конструкторов

- Создайте класс Book с несколькими конструкторами:
- пустой,



- с названием и автором,
- с названием, автором и годом издания.

14. Создайте классы Transport (базовый), Car, Bike (наследники). Реализуйте общий метод *move()* и переопределите его в наследниках.

15. Создайте массив объектов базового класса Animal, содержащий объекты Dog, Cat, Cow. Реализуйте метод *sound()*, который вызывается полиморфно.

16. Создайте абстрактный класс Shape с абстрактным методом *getArea()*. Реализуйте наследников Circle и Rectangle.

17. Создайте интерфейс ILogger с методом *log(message)*. Реализуйте два класса: ConsoleLogger  
FileLogger

которые по-разному обрабатывают сообщения.

18. Создайте класс Library, который хранит список книг и предоставляет методы:

- добавить книгу,
- удалить книгу,
- найти книгу по названию.

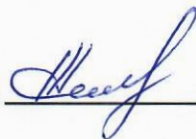
19. Реализуйте собственное исключение InvalidAgeException, которое выбрасывается при попытке задать возраст меньше 0 или больше 120.

20. Разработайте упрощённую систему Онлайн-магазина, включающую классы:

- Product,
- Customer,
- Order,
- OrderItem.
- Реализуйте:
  - добавление товаров в заказ;
  - расчёт итоговой стоимости;
  - вывод информации о заказе.

Директор института

энергетики и цифровых технологий



Тойгожинова А.Ж.

Заведующий кафедрой

информационно-коммуникационных технологий



Касымова Д.Т.