

АО «ALT университет имени М. Тынышпаева»

Институт энергетики и цифровых технологий

Кафедра информационно-коммуникационных технологий

УТВЕРЖДАЮ
Председатель УС
Жармагамбетова М.С.
22г.
Решение УС от «22» 2025г. пр. № 3



**ПРОГРАММА
АТТЕСТАЦИОННОГО (КОМПЛЕКСНОГО) ЭКЗАМЕНА**

по образовательной программе «6B06118 Программная инженерия»

Алматы, 2025

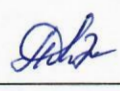
Программа Аттестационного (комплексного) экзамена по базовой и профилирующим дисциплинам образовательной программы «6В06118 Программная инженерия» составлена в соответствии с ГОСО, утвержденным Приказом Министра науки и высшего образования РК № 2 от 20 июля 2022г., Типовых правил деятельности организаций образования соответствующих типов и видов, утвержденных Приказом Министра образования и науки РК № 595 от 30 октября 2018 года, в редакции приказа и.о Министра образования и науки РК от 29.12.2021 № 614, образовательной программы 6В06118 Программная инженерия», рабочих учебных программ дисциплин (РУПД).

Форма обучения: очная.

Программа рассмотрена и обсуждена на заседании кафедры информационно-коммуникационных технологий.

Протокол № 2 от «22» октября 2025 г.

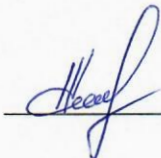
Заведующий кафедрой

информационно-коммуникационных технологий  Касымова Д.Т.

Программа рассмотрена и одобрена на заседании УМБ института энергетики и цифровых технологий.

Протокол № 2 от «29» октября 2025 г.

Председатель УМБ института

энергетики и цифровых технологий  Тойгожинова А.Ж.

Программа рассмотрена и рекомендована на утверждение УС на заседании УМС.
Протокол № 2 от 20.11.2025 г.

Председатель УМС

 Абдрешов Ш.А.

Содержание

1. Цель Аттестационного (комплексного) экзамена	4
2. Регламент проведения Аттестационного (комплексного) экзамена	4
3. Критерии и показатели оценки знаний обучающихся	5
4. Содержание Аттестационного (комплексного) экзамена	7
5. Рекомендуемая литература	11

1. Цель Аттестационного (комплексного) экзамена по образовательной программе

Целью Аттестационного (комплексного) экзамена по базовой и профилирующим дисциплинам образовательной программы «6B06118 Программная инженерия» является определение степени, соответствия уровня подготовленности выпускников требованиям ГОСО бакалавриата, утвержденными Приказом Министра науки и высшего образования РК № 2 от 20 июля 2022г., Типовых правил деятельности организаций образования соответствующих типов и видов, утвержденных Приказом Министра образования и науки РК № 595 от 30 октября 2018 года, в редакции приказа и.о Министра образования и науки РК от 29.12.2021 № 614, образовательной программы «6B06118 Программная инженерия» и рабочих учебных программ дисциплин (РУПД).

При проведении Аттестационного (комплексного) экзамена проверяются как теоретические знания, так и практические навыки выпускника в соответствии с образовательной программой «6B06118 Программная инженерия».

2. Регламент проведения Аттестационного (комплексного) экзамена по образовательной программе

Аттестационный (комплексный) экзамен в качестве итоговой аттестации проводится в соответствии с академическим календарем и согласно графику учебного процесса на текущий учебный год.

К экзамену допускаются лица, завершившие полный курс обучения и успешно прошедшие все предшествующие аттестационные испытания, предусмотренные учебным планом. Для проведения итоговой аттестации обучающихся по образовательной программе «6B06118 Программная инженерия» создается Аттестационная комиссия (далее – АК).

Председатель АК и персональный состав комиссии утверждается Председатель Правления-Ректором АЛТ университет.

В компетенцию Аттестационной комиссии входят:

- проверка уровня соответствия теоретической и практической подготовки выпускаемых кадров, установленных требованиям образовательных программ;
- присуждение выпускнику степени бакалавра по соответствующей образовательной программе;
- разработка предложений, направленных на дальнейшее улучшение качества подготовки кадров.

Программа Аттестационного (комплексного) экзамена по базовой и профилирующим дисциплинам доводится до сведения обучающихся не позднее, чем за месяц до предполагаемой даты проведения экзамена и публикуется на сайте Университета.

Обсуждение и окончательное оценивание ответов обучающихся, АК проводит на закрытом заседании, определяя итоговую оценку – в балльно-рейтинговом значении (таблица 1 – Критерии и показатели оценки знаний обучающихся).

Результаты Аттестационного (комплексного) экзамена доводятся до обучающихся в день сдачи экзамена.

Повторная сдача Аттестационного (комплексного) экзамена с целью повышения положительной оценки не допускается.

Пересдача Аттестационного (комплексного) экзамена, получившим оценку «неудовлетворительно», в данный период итоговой аттестации не разрешается.

Обучающийся, получивший по итоговой аттестации оценку «неудовлетворительно», отчисляется из Университета приказом Председателя Правления-Ректора как «не выполнивший требования образовательной программы: не сдавший Аттестационный (комплексный) экзамен».

Обучающемуся, прошедшему итоговую аттестацию и подтвердившему освоение образовательной программы, решением Аттестационной комиссии присуждается степень «бакалавр» и выдается диплом с приложением.

3. Критерии и показатели оценки знаний обучающихся

№	Оценка в буквенной системе	Цифровой эквивалент баллов	%-ное содержание	Оценка по традиционной системе	Показатели критериев оценивания
1	2	3	4	5	6
1	A	4,0	95-100	отлично	1.Полностью, правильно с учётом современной теории излагает материал; 2.Показывает дополнительные знания; 3.Теоретические знания увязывает с практикой; 4.Свободно владеет терминологией; 5.Устанавливает причинно-следственные связи; 6.Умеет делать прогноз; 7.Уверенно отвечает на дополнительные вопросы.
2	A-	3,67	90-94		1.Правильно с учётом современной теории излагает материал; 2.Отвечает полностью, самостоятельно делает выводы и обобщения; 3.Хорошо знает терминологию; 4.Устанавливает причинно-следственные связи. 5.Полностью отвечает на дополнительные вопросы.
3	B+	3,33	85-89	хорошо	1.Хорошо излагает материал; 2.Отвечает полностью, самостоятельно делает выводы и обобщения; 3.Владеет терминологией; 4.Владеет логикой. 5.Отвечает на дополнительные вопросы
4	B	3,0	80-84		1.Знает основной материал; 2.Обоснованно приводит примеры; 3.Делает обобщения и выводы; 4.Допускает неточности в терминологии, логике изложения. 5.Отвечает на дополнительные вопросы
5	B-	2,67	75-79		1.Знает основной материал, но отвечает сбивчиво без логики; 2.Допускает неточности при пользовании терминами; 3.При ответах на дополнительные вопросы допускает ошибки.

6	C+	2,33	70-74		1.Имеет только основы теоретических знаний; 2.Не умеет делать выводов и обобщений; 3.Не пользуется терминологией; 4.Отвечает на дополнительные и уточняющие вопросы.
7	C	2,0	65-69	удовлетворительно	1.Имеет неполные знания основного материала; 2.Не владеет логикой; 3.Ответ носит фрагментарный характер; 4.На дополнительные вопросы отвечает не полностью.
8	C-	1,67	60-64		1.Имеет неполные знания основного материала; 2.Допускает неточности, не умеет делать выводы, обобщения; 3.Не владеет логикой; 4.Допускает ошибки при ответах на дополнительные вопросы.
9	D+	1,33	55-59		1.Не знает значительной части материала; 2.Слабо ориентируется в материале; 3.Не владеет логикой ответа на вопрос; 4.Затрудняется отвечать на дополнительные вопросы.
10	D	1,0	50-54		1.Знание и понимание основ теории с допущением существенных неточностей, некоторая поверхностность, снижающая понимание, но способность с помощью наводящих вопросов преподавателя приходить к правильному суждению.
11	F	0	0-49	неудовлетворительно	1.Отрывочные знания, изложение их без понимания смысла. 2.Дополнительные вопросы членов комиссии только затрудняют мысль, формально заученные положения теории.

4. Содержание Аттестационного (комплексного) экзамена по образовательной программе «6В06118 Программная инженерия»

Тематика экзаменационных вопросов соответствует избранным разделам из рабочих учебных программ (силлабусов) циклов по базовой и профилирующим дисциплинам:

- 4.1. Профилирующая дисциплина - **«Инструментальные средства разработки программ»**
- 4.2. Профилирующая дисциплина – **«Основы компьютерного моделирования»**
- 4.3. Базовая дисциплина - **«Объектно-ориентированное программирование»**
- 4.4. Задачи

4.1. Профилирующая дисциплина - «ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММ»

4.1.1. Введение в инструментальные средства разработки ПО

Понятие инструментальных средств разработки. Классификация инструментов (редакторы, IDE, системы контроля версий, тестирование и др.). Роль инструментов в жизненном цикле ПО. Автоматизация процессов разработки. Обзор современных тенденций (DevOps, CI/CD, AI-инструменты).

4.1.2. Среда разработки программ (IDE)

Архитектура и компоненты IDE. Редактор кода: подсветка синтаксиса, автодополнение, рефакторинг. Консоль управления проектом. Инструменты сборки и запуска. Плагины и расширения. Сравнение популярных IDE (VS Code, PyCharm, IntelliJ IDEA, Eclipse). Понятие версионирования. Центральные и распределённые системы контроля версий. Основные команды Git (commit, push, pull, merge). Работа с ветками (branching). Разрешение конфликтов. Репозитории GitHub/GitLab/Bitbucket. Командная работа и workflow (GitFlow)

4.1.3. Инструменты отладки программ

Типы ошибок в программировании (логические, синтаксические, runtime). Debugger: точки останова, просмотр переменных, пошаговое выполнение. Логирование и мониторинг. Инструменты профилирования (CPU, память). Поиск утечек памяти. Встроенные отладчики в IDE

4.1.4. Инструменты автоматизации сборки ПО

Понятие сборки программного проекта. Скрипты и процессы сборки. Make, CMake в C/C++. Maven и Gradle для Java. NPM/Yarn для JavaScript-проектов. Контейнеризация сборок. Управление зависимостями. Виды тестирования (unit, integration, UI, нагрузочное). Фреймворки автоматизированного тестирования (JUnit, PyTest, Selenium). Тестовые сценарии и покрытие тестами. Mocking и тестирование зависимостей. CI-интеграция тестов. Отчётность и инструменты анализа тестов.

4.1.5. Документирование программных продуктов

Требования к документации (техническая, пользовательская, API). Инструменты генерации документации (Sphinx, Javadoc, Swagger). Поддержка актуальности документации. Автоматизация документации. Форматы: Markdown, HTML, XML, PDF.

CASE-средства и моделирование программных систем. Понятие CASE-систем. UML-диаграммы (классов, последовательностей, вариантов использования). ER-диаграммы для проектирования БД. Автоматическая генерация кода. Rational Rose, Enterprise Architect, Draw.io. Архитектурное моделирование.

4.1.6. DevOps-инструменты и CI/CD

Концепции DevOps и культура сотрудничества. Инструменты CI/CD (Jenkins, GitLab CI, GitHub Actions). Контейнеризация: Docker, образы, Dockerfile. Оркестрация: Kubernetes (обзор). Автоматизация развёртывания приложений. Мониторинг (Prometheus, Grafana). Инструменты анализа и контроля качества кода. Статический анализ кода. Линтеры (ESLint, Pylint, Checkstyle). Инструменты поиска уязвимостей (SonarQube). Code

Review и стандарты кодирования. Метрики качества ПО. Автоматизация проверки качества.

4.2 Профилирующая дисциплина - ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

4.2.1. Введение в компьютерное моделирование. Понятие модели и моделирования

Классификация моделей: математические, имитационные, статистические. Этапы разработки модели. Роль Python в научных расчётах и моделировании. Основы Python для моделирования. Переменные, типы данных. Условные операторы и циклы. Функции и модули для вычислений. Работа с файлами (данные для модели).

4.2.2. Математическое моделирование на Python. Линейные и нелинейные функции. Системы уравнений и их решения (SymPy). Построение графиков функций (Matplotlib). Численные методы: приближённые решения. Численные методы в компьютерном моделировании. Метод Эйлера. Метод Рунге–Кутты. Приближённое интегрирование. Численные решения дифференциальных уравнений (SciPy).

4.2.3. Статистическое моделирование. Генерация случайных величин (NumPy). Вероятностные распределения. Монте-Карло моделирование. Анализ статистических данных. Имитационное моделирование. Дискретные события. Модели очередей. Симуляция процессов с использованием библиотек: SimPy. Построение временных диаграмм. Моделирование физических процессов. Модели движения: скорость, ускорение, траектория. Колебательные системы (математический маятник). Модели теплопередачи. Численное решение PDE (уравнений в частных производных).

4.2.4. Пространственное моделирование и визуализация. Генерация данных для 2D/3D моделей. Matplotlib 3D, Plotly. Анимация процессов (FuncAnimation). Моделирование поверхностей и полей. Оптимизационные модели. Линейная оптимизация (SciPy.optimize). Поиск минимума функции. Моделирование процессов оптимизации. Применение оптимизации в экономике, логистике, ИТ

4.2.5. Машинное обучение как инструмент моделирования. Регрессия как модель зависимости. Классификация и принятие решений. Построение моделей прогнозирования (sklearn). Оценка точности моделей. Моделирование сложных систем. Мультиагентное моделирование. Параллельные процессы. Использование Python-фреймворков: Mesa. Анализ поведения систем. Проектирование собственной модели. Постановка задачи. Сбор данных. Разработка модели в Jupyter Notebook. Валидация и визуализация результатов.

4.3. Базовая дисциплина - ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

4.3.1. Введение в ООП. Понятие объекта и класса. История и назначение ООП. Основные принципы: инкапсуляция, наследование, полиморфизм, абстракция. Преимущества ООП в разработке ПО

4.3.2. Классы и объекты в Python. Создание класса (class). Экземпляры (объекты). Атрибуты класса и объекта. Метод `__init__()` (конструктор). Примеры моделирования реальных объектов

4.3.3. Инкапсуляция и методы доступа. Публичные, защищённые и приватные атрибуты. Геттеры и сеттеры. Свойства (`@property`). Соккрытие данных, контроль доступа

4.3.4. Наследование. Базовый и дочерний классы. Переопределение методов. `super()` и расширение функциональности. Множественное наследование. Полиморфизм. Переопределение методов. Понятие интерфейса. Одинаковый интерфейс — разное поведение. Примеры: животные, транспорт, банковские операции.

4.3.5. Абстракция. Абстрактные классы. Модуль abc и ABCMeta. Абстрактные методы. Применение абстракции в архитектуре программ. Магические методы (dunder methods), `__str__`, `__repr__`. Арифметические методы: `__add__`, `__sub__`, Сравнения: `__eq__`, `__lt__`. Итерация: `__iter__`, `__next__`. Использование для моделирования логики объектов

4.3.6. Работа с коллекциями и ООП. Пользовательские контейнеры. Итераторы и генераторы. Создание собственных структур данных: стек, очередь, список. Исключения и обработка ошибок. try / except. Создание собственных исключений. Исключения в объектах. Логирирование ошибок

4.3.7. Композиция и агрегация. Отношение «часть–целое». Отличие от наследования. Построение сложных моделей: автомобиль, университет, система бронирования. Паттерны проектирования. Работа с файлами и объектами. ООП и графические интерфейсы. Объектно-ориентированное моделирование

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.

5. Рекомендуемая литература

Основная литература.

- 5.1. Гамильтон Р. *Visual Studio 2022: Полное руководство разработчика*. — М.: Вильямс, 2022.
- 5.2. Лутц М. *Изучаем Python*. 5-е изд. — СПб.: Питер, 2021.
- 5.3. Сьерра К., Бейтс Б. *Head First: Программирование*. — O'Reilly Media, 2020.
- 5.4. Фаулер М. *Рефакторинг. Улучшение существующего кода*. — СПб.: Питер, 2020.
- 5.5. Буч Г. *Объектно-ориентированный анализ и проектирование с примерами приложений*. — М.: Вильямс, 2019.
- 5.6. Лутц М. *Программирование на Python*. — СПб.: Питер, 2022.
- 5.7. Шилдт Г. *Java. Полное руководство*. — М.: Вильямс, 2021.

- 5.8. Страуструп Б. *Язык программирования C++*. — М.: БХВ-Петербург, 2020.
- 5.9. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. *Паттерны проектирования*. — М.: Вильямс, 2021.
- 5.10. Буч Г. *ООП: принципы и практика*. — М.: Бином, 2020.
- 5.11. Федосеев П. *Компьютерное моделирование: учебник*. — М.: Юрайт, 2021.
- 5.12. Самарский А.А., Михайлов А.П. *Математическое моделирование*. — М.: Наука, 2020.
- 5.13. Higham D., Higham N. *Mathematical Modelling Using Python*. — SIAM, 2019.
- 5.14. Марон И. *Численные методы*. — М.: Физматлит, 2022.
- 5.15. Златопольский Д. *MATLAB и моделирование сложных систем*. — М.: ДМК, 2021.

Дополнительная литература

- 5.1. McConnell S. *Code Complete*. 2nd Ed. - Microsoft Press, 2019.
- 5.2. Sommerville I. *Software Engineering*. 10th Ed. - Pearson, 2020.
- 5.3. Мартин Р. *Чистая архитектура*. — М.: Питер, 2020.
- 5.4. Канер С., Фолк Дж. *Тестирование программного обеспечения*. — СПб.: Питер, 2021.
- 5.5. Martelli A., Raven W. *Python Cookbook*. -O'Reilly, 2019.
- 5.6. Eckel B. *Thinking in Java*. 4th Ed. - Pearson, 2019.
- 5.7. Bloch J. *Effective Java*. 3rd Ed. - Addison-Wesley, 2020.
- 5.8. Grady Booch. *Object-Oriented Design with Applications*. — Addison-Wesley, 2021.
- 5.9. Liang Y. *Introduction to Java Programming and Data Structures*. — Pearson, 2020.
- 5.10. Muller J. *Computational Modeling and Simulation in Engineering*. — CRC Press, 2019.
- 5.11. Griva I., Nash S. *Linear and Nonlinear Optimization*. — SIAM, 2020.
- 5.12. Guttag J. *Introduction to Computation and Programming Using Python*. — MIT Press, 2021.
- 5.13. Burden R., Faires D. *Numerical Analysis*. — Cengage Learning, 2020.
- 5.14. Allen M. *Python for Computational Science and Engineering*. — Apress, 2021.