Федеральное государственное бюджетное образовательное учреждение высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»

Институт экономики, математики и информационных технологий

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат:
02EAC3960059B0ABB44AB6DF5620B43941
Владелец: Маруев Сергей Александрович
Действителен: с 09.08.2023 до 28.04.2038

«УТВЕРЖДАЮ»

Директор Института ЭМИТ
РАНХиГС

А.Д. Радыгин

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
повышения квалификации

Программирование

Москва, 2026

Разработчик:

С.И. Никишов
К.т.н., д.э.н., доцент
Заведующий кафедрой
Системной и программной
инженерии
ФИТАД ИЭМИТ РАНХиГС

К.А. Алексеев
Главный специалист Института
ЭМИТ РАНХиГС

Руководитель программы:

С.А. Маруев
К.т.н., доцент
Заведующий кафедрой
Системного анализа и информатики
ФИТАД ИЭМИТ РАНХиГС

Дополнительная профессиональная программа повышения квалификации рассмотрена и одобрена на заседании ученого совета Института ЭМИТ и рекомендована к утверждению и реализации «18» февраля 2026 г., протокол № 03-25/26.

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ 4
1.1. Цель реализации программы 4
1.2. Нормативные правовые акты 4
1.3. Планируемые результаты обучения 5
1.4. Категория слушателей 6
1.5. Формы и технологии обучения 6
1.6. Период обучения, срок освоения и режим занятий 6
1.7. Документ о квалификации 6
2. СОДЕРЖАНИЕ ПРОГРАММЫ 7
2.1. Календарный учебный график 7
2.2. Учебный план 1
2.3. Содержание программы по темам 3
3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ ПРОГРАММЫ 4
3.1. Материально-техническое и программное обеспечение реализации программы 4
3.2. Учебно-методическое и информационное обеспечение программы 4
4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ 6
5. ИНДИКАТОРЫ СФОРМИРОВАННЫХ КОМПЕТЕНЦИЙ……..11

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель реализации программы

Целью реализации программы является совершенствование компетенций и получение новой компетенции, необходимой для профессиональной деятельности, повышение профессионального уровня в рамках имеющейся научной квалификации, а также получение слушателями профессиональных компетенций, необходимых для успешного выполнения задач, связанных с разработкой программного обеспечения на языке программирования.

1.2. Нормативные правовые акты

Дополнительная профессиональная программа повышения квалификации разработана на основании следующих нормативных документов:

1. Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 25.12.2023) "Об образовании в Российской Федерации" (с изм. и доп., вступ. в силу с 01.01.2024).
2. Приказ Минобрнауки России от 24.03.2025 N 266 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам" (Зарегистрировано в Минюсте России 22.04.2025 N 81928)
3. Профессиональный стандарт «Программист», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022г. №424н.
4. «Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов» (утв. Минобрнауки России 22.01.2015 № ДЛ-1/05вн).
5. Образовательный стандарт высшего образования федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» направление подготовки 01.03.02 «Прикладная математика и информатика» (уровень бакалавриата) утвержден приказом Минобрнауки №9 от 10 января 2018 г. (с изменениями).
6. Приказ РАНХиГС от 19.04.2019г. №02-461 «Об утверждении локальных нормативных актов РАНХиГС по дополнительному профессиональному образованию» (п. 3 Порядок реализации дополнительных профессиональных программ в РАНХиГС)
7. Приказ РАНХиГС от 02 декабря 2025 года №02-02669/001 об утверждении «Порядка разработки и утверждения в Академии дополнительных профессиональных программ - программ повышения квалификации, программ профессиональной переподготовки».
8. Приказ от 13 января 2026 года N 02-00010/001 "Об утверждении Правил приема на обучение по дополнительным профессиональным программам в Академию"
9. Приказ от 13 января 2026 года №02-00009/001 «Об утверждении Положения об итоговой аттестации слушателей дополнительных профессиональных программ в Академии».
10. Приказ Федеральной службы по надзору в сере образования и науки от 4 августа 2023г. №1493 об утверждении требований к структуре официального сайта образовательной организации в информационно-телекоммуникационной сети «Интернет» и формату представления информации (в ред. Приказа Рособрнадзора от 03.07.2025№1353)

11. Единый квалификационный справочник должностей руководителей, специалистов и других служащих (ЕКС, ЕКСД), 2026. Единый квалификационный справочник должностей руководителей, специалистов и других служащих, «Общеотраслевые квалификационные характеристики должностей работников, занятых на предприятиях, в учреждениях и организациях» и «Квалификационные характеристики должностей работников, занятых в научно-исследовательских учреждениях, конструкторских, технологических, проектных и изыскательских организациях» (Постановление Минтруда РФ от 21.08.1998 № 37 ред. от 27.03.2018г.).

12. Общероссийский классификатор видов экономической деятельности (ОКВЭД2) ОК 029-2014 (Приказ Росстандарта от 10.11.2015 № 1745-ст).

13. Общероссийский классификатор профессий рабочих, должностей служащих и тарифных разрядов ОК 016-2025(принят и введен в действие) Приказом Федерального агентства по техническому регулированию и метрологии от 16 мая 2025 г. N 423-ст (взамен Общероссийского классификатора профессий рабочих, должностей служащих и тарифных разрядов ОК 016-94)

1.3. Планируемые результаты обучения

Таблица 1

Перечень профессиональных компетенций в рамках имеющейся квалификации и профессиональных компетенций, планируемых к освоению (результаты обучения)

Виды деятельности	**Общепрофессиональные/ профессиональные компетенции ОПК, ПК или трудовые функции (ПСК и СК)** ***(формируются и (или) совершенствуются)***	**Практический опыт**	**Знания**	**Умения**
1	2	**3**	4	5
ВД Разработка и тестирование программного обеспечения на языке Python Разработка компьютерного программного обеспечения	ПСК-1. Оформление программного кода в соответствии с установленными требованиями (A/03.3)*	Разработка простых скриптов и утилит. Решение прикладных задач с применением базовых возможностей Python. Обработка данных из файлов.	Основные синтаксические конструкции языка Python. Принципы структурного программирования. Основные типы данных, структуры данных, файловые форматы JSON и CSV.	Писать программы на языке Python, используя переменные, типы данных, условия и функции. Использовать структуры данных для хранения и обработки информации. Читать и записывать файлы в формате TXT, JSON, CSV.

	ПСК-2. Проверка и отладка программного кода (A/05.3)*	Создание и отладка серверных API с использованием FastAPI. Реализация CRUD-операций и взаимодействие с БД. Интеграция внешних API в приложение.	Основы REST API, библиотека requests, взаимодействие по протоколу HTTP. Основы работы с базами данных и ORM (SQLAlchemy). Основы объектно-ориентированного программирования.	Разрабатывать backend-приложения на FastAPI. Использовать Pydantic для валидации данных. Подключать и использовать базы данных (SQLite/PostgreSQL) через ORM. Использовать requests для интеграции с внешними API.
	ПСК-3. Разработка тестовых наборов данных для проверки работоспособности компьютерного программного обеспечения (B/02.4)*	Проведение отладки программного кода. Реализация и запуск тестов. Использование архитектурных паттернов в небольших проектах. Рефакторинг кода в соответствии со стандартами.	Основы модульного тестирования и библиотеки pytest. Основы логирования. Принципы организации архитектуры backend-приложения. Обработка сключений в Python.	Писать модульные тесты, выявлять и устранять ошибки в коде. Использовать логирование для отладки. Разрабатывать архитектурно организованные проекты. Применять принципы чистого кода.

*Сформирована на базе проф. стандартом «Программист» Приказ от 20.07.2022г. №424н

1.4. Категория слушателей

К освоению программы допускаются лица, имеющие высшее образование или среднее профессиональное образование.

1.5. Формы и технологии обучения

Форма и технологии обучения – очно-заочная (с применением дистанционных образовательных технологий).

1.6. Период обучения, срок освоения и режим занятий

Периоды обучения составляют: 9 недель (см. табл. 2.1), 6 недель (см. табл. 2.2) и 4 недели (см. табл. 2.3).

Трудоемкость программы составляет 108 академических часов, в том числе 72 часа контактной работы, 36 часа самостоятельной работы и 4 академических часа, отведенных на итоговую аттестацию с применением дистанционных образовательных технологий.

Режим занятий: с понедельника по субботу (в соответствии с расписанием), но не менее 2 дней в неделю. Более точное время проведения занятий корректируется расписанием по согласованию с преподавателями.

1.7. Документ о квалификации

Удостоверение о повышении квалификации федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации».

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Календарный учебный график

Таблица 2.1

Календарный учебный график

Период обучения – 9 недель								
1 неделя	2 неделя	3 неделя	4 неделя	5 неделя	6 неделя	7 неделя	8 неделя	9 неделя
УЗ ДОТ	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ, ИА ДОТ*

Таблица 2.2

Календарный учебный график

Период обучения – 6 недель					
1 неделя	2 неделя	3 неделя	4 неделя	5 неделя	6 неделя
УЗ ДОТ	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ, ИА ДОТ*

Таблица 2.3

Календарный учебный график

Период обучения – 4 недели			
1 неделя	2 неделя	3 неделя	4 неделя
УЗ ДОТ	*УЗ ДОТ*	*УЗ ДОТ*	*УЗ ДОТ, ИА ДОТ*

Примечание: УЗ ДОТ– учебные занятия с применением дистанционных образовательных технологий, ИА ДОТ – итоговая аттестация с применением дистанционных образовательных технологий.

2.2. Учебный план

Таблица 3

Учебный план

№ п/п	Наименование дисциплины (модуля), практики (стажировки)	Общая трудоемкость, час	Контактная работа, час.					Самостоятельная работа, час	Контактная работа (с применением дистанционных образовательных технологий, электронного обучения), час.					Самостоятельная работа, час	Текущий контроль успеваемости	Промежуточная аттестация	Итоговая аттестация (вид/час)	Код компетенции
			Всего	В том числе					Всего	В том числе								
				Лекции / в интерактивной форме	Лабораторные занятия (практикум) /в интерактивной форме	Практические (семинарские) занятия /в интерактивной форме	Контактная самостоятельная работа, час			Лекции/ в интерактивной форме	Практические (семинарские/лабораторные) занятия /в интерактивной форме Практические	Контактная самостоятельная работа, час	Индивидуальные и групповые консультации					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1.	**Программирование**	104	-	-	-	-	-	-	72	40	32	-	-	32	-	-	-	ПСК-1 ПСК-2 ПСК-3
1.1	Введение в Python. Синтаксис, переменные, типы данных	12	-	-	-	-	-	-	8	4	4	-	-	4	-	-	-	ПСК-1 ПСК-2
1.2	Условные конструкции и функции	10	-	-	-	-	-	-	6	4	2	-	-	4	-	-	-	ПСК-1 ПСК-2
1.3	Структуры данных: списки, словари, множества, кортежи	14	-	-	-	-	-	-	10	4	6	-	-	4	-	-	-	ПСК-1 ПСК-2
1.4	Работа с файлами и модулями. Управление пакетами	10	-	-	-	-	-	-	6	4	2	-	-	4	-	-	-	ПСК-1

																		ПСК-2
1.5	Основы ООП в Python	8	-	-	-	-	-	-	6	4	2	-	-	2	-	-	-	ПСК-2 ПСК-3
1.6	Работа с API: библиотека requests, парсинг данных	12	-	-	-	-	-	-	8	4	4	-	-	4	-	-	-	ПСК-1
1.7	Основы работы с базами данных и SQLAlchemy	8	-	-	-	-	-	-	6	4	2	-	-	2	-	-	-	ПСК-1
1.8	Разработка приложений на FastAPI	12	-	-	-	-	-	-	8	4	4	-	-	4	-	-	-	ПСК-1
1.9	Тестирование и отладка Python-кода.	10	-	-	-	-	-	-	8	4	4	-	-	2	-	-	-	ПСК-2 ПСК-3
1.10	Архитектура приложений	8	-	-	-	-	-	-	6	4	2	-	-	2	-	-	-	ПСК-1
	Итого:	**104**	-	-	-	-	-	-	**72**	**40**	**32**	-	-	**32**	-	-	-	
	Итоговая аттестация	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**4** З(Д)	-
	Всего:	**108**	-	-	-	-	-	-	**72**	**40**	**32**	-	-	**32**	-	-	**4**	-

З (Д) – зачет в форме тестирования с применением ДОТ

2.3. Содержание программы по темам

Таблица 4

Содержание программы по темам

Номер темы и её наименование	Содержание темы
Тема 1.1. Введение в Python. Синтаксис, переменные, типы данных	Что такое язык программирования? Особенности Python и его применение. Установка и настройка среды разработки (IDE, интерпретатор). Первые программы: вывод данных. Переменные и типы данных (int, float, str, bool). Основные операции с числами и строками. Ввод данных с клавиатуры. Ошибки в коде и их исправление.
Тема 1.2. Условные конструкции и функции	Понятие логики программ. Операторы сравнения и логические операторы. Условные конструкции if-elif-else. Вложенные условия. Что такое функция? Объявление и вызов функций. Аргументы и параметры. Возвращаемое значение. Область видимости переменных. Практика: написание простых функций для автоматизации задач.
Тема 1.3. Структуры данных: списки, словари, множества, кортежи	Понятие коллекций данных. Работа со списками: создание, индексация, методы list. Словари: ключ-значение, методы dict. Множества и их особенности. Кортежи как неизменяемые списки. Операции над коллекциями. Вложенные структуры данных. Практика обработки данных с использованием структур.
Тема 1.4. Работа с файлами и модулями. Управление пакетами	Чтение и запись файлов. Работа с текстовыми файлами. Контекстный менеджер. Основы работы с форматами JSON и CSV. Понятие модулей и импортов. Создание собственных модулей. Управление зависимостями: pip, виртуальные окружения (venv). Установка сторонних библиотек.
Тема 1.5. Основы ООП в Python	Что такое объектно-ориентированное программирование? Понятие классов и объектов. Атрибуты и методы. Конструкторы. Наследование и полиморфизм. Инкапсуляция. Примеры использования ООП в реальных проектах. Практика проектирования классов для решения задач.
Тема 1.6. Работа с API: библиотека requests, парсинг данных	Понятие API и HTTP-запросов. Метод GET и POST. Работа с библиотекой requests. Отправка запросов и обработка ответов. Работа с JSON-ответами. Парсинг данных из внешних сервисов. Практика интеграции с публичными API. Ошибки при работе с API и их обработка.
Тема 1.7. Основы работы с базами данных и SQLAlchemy	Введение в реляционные базы данных. Основы SQL: SELECT, INSERT, UPDATE, DELETE. Работа с SQLite/PostgreSQL. Подключение Python-приложения к БД. Введение в ORM. Использование SQLAlchemy: создание моделей, выполнение запросов. Практика построения CRUD-операций через ORM.
Тема 1.8. Разработка приложений на FastAPI	Что такое backend и REST API? Установка FastAPI и Uvicorn. Создание первого маршрута (endpoint). Обработка GET и POST запросов. Работа с параметрами и телом запроса. Ответы сервера и коды статусов. Использование Pydantic для валидации данных. Подключение к БД через FastAPI. Запуск и тестирование API.
Тема 1.9. Тестирование и отладка Python-кода	Понятие тестирования кода. Зачем нужны тесты? Введение в модульное тестирование. Использование библиотеки pytest: написание базовых тестов для функций и классов. Организация структуры тестов в проекте. Основы логирования с помощью библиотек logging. Практика выявления и исправления ошибок. Использование инструментов IDE для отладки. Обработка исключений в Python (try/except). Разбор типичных ошибок при разработке backend-приложений.
Тема 1.10. Архитектура приложений	Что такое архитектура приложения и зачем она нужна? Принципы организации кода в реальных проектах. Разделение логики: routers, models, schemas, services, utils. Использование Pydantic для валидации данных. Dependency Injection в FastAPI. Работа с конфигурационными файлами (.env, библиотека pydantic-settings). Основы чистого кода и паттерны проектирования для начинающих. Типовая структура backend-приложения на FastAPI. Практика: рефакторинг небольшого проекта, приведение его к архитектурным стандартам.

3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ ПРОГРАММЫ

3.1. Материально-техническое и программное обеспечение реализации программы

Для проведения лекций, лабораторных и практических занятий, а также итоговой аттестации необходим персональный компьютер или ноутбук, оснащённый микрофоном и наушниками, с предустановленной программой для видеоконференций — Яндекс.Телемост, а также современным браузером (рекомендуется Яндекс браузер) и стабильным выходом в интернет на скорости от 10 Мбит/с.

Программное обеспечение, необходимое для реализации программы:

1) Microsoft Excel — табличный процессор, используемый для обработки и анализа данных, проведения вычислений, построения сводных таблиц, а также визуализации информации. Подойдёт версия Microsoft Excel 2016 или более новая, либо онлайн-версия Office 365.
2) Python — высокоуровневый язык программирования общего назначения с динамической типизацией и автоматическим управлением памятью. Отличается лаконичным синтаксисом, масштабируемостью, простотой в изучении и богатой экосистемой библиотек для анализа данных.
3) Anaconda — популярный дистрибутив Python, предназначенный для научных и аналитических задач. Включает множество библиотек и инструментов для анализа, машинного обучения и научных вычислений, облегчает установку и управление зависимостями.
4) Jupyter Notebook — интерактивная среда для разработки программ, которая позволяет создавать и документировать исполняемые аналитические проекты с пояснениями, кодом и результатами вычислений. Доступен как локально, так и через облачные сервисы.
5) Google Colab — бесплатная облачная среда для запуска и обмена Python-кодом, позволяющая работать с Jupyter-ноутбуками прямо в браузере без локальной установки специального ПО.
6) NumPy — библиотека Python для работы с многомерными массивами и выполнения математических вычислений. Является основой для большинства пакетов аналитики на Python.
7) Pandas — библиотека для обработки и анализа табличных данных, предоставляющая расширенные средства для фильтрации, агрегации, визуализации и предобработки данных.
8) Matplotlib и Seaborn — библиотеки для визуализации данных и построения графиков различной сложности непосредственно в процессе аналитики.
9) Beautiful Soup — Python-библиотека для парсинга и извлечения данных из HTML и XML-документов. Применяется при необходимости сбора данных из открытых интернет-источников.
10) Yandex DataLens — облачная BI-платформа, предоставляющая инструменты для создания интерактивных дашбордов, визуализации структурированных данных, построения и публикации аналитических отчётов. Для подключения к сервису необходим аккаунт Яндекса и доступ в интернет.
11) Дополнительно для совместной работы и обмена файлами может использоваться сервис Яндекс.Диск. Для оформления аналитических заключений и презентаций слушателям рекомендуется стандартное офисное ПО (Word, PowerPoint или аналоги).
12) DBeaver — это клиентское приложение SQL и инструмент администрирования базы данных.

3.2. Учебно-методическое и информационное обеспечение программы

Положение об организации самостоятельной работы студентов федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (в ред. приказа РАНХиГС от 11.05.2016 г. № 01-2211).
https://www.ranepa.ru/images/docs/prikazy-ranhigs/Pologenie_o_samostoyatelnoi_rabote.pdf

Для успешного освоения программы слушателю необходимо регулярно заниматься самостоятельной работой.

В процессе самостоятельной работы стоит уделить особое внимание в подготовке к тестированию, типовых заданий по дисциплине, в решении слушателям необходимо

обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы.

Самостоятельная работа заключается в чтении конспектов лекций, изучении разделов курса по материалам учебника, учебными пособиями, самостоятельном изучении научных статей по темам курса, справочной и популярной литературой, материалами периодических изданий и Интернета, выполнении домашних заданий, решении дополнительных задач.

Самостоятельная работа со статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у слушателя компьютерную грамотность и культуру использования цифровой информации.

Подготовке к тестированию и выполнению практического задания, а также более глубокому раскрытию вопросов, способствует самостоятельное знакомство с дополнительной литературой.

Примеры заданий для самостоятельной работы:

Задание 1

Работа с API и парсинг данных.

Что нужно сделать:

Воспользуйтесь публичным API - https://jsonplaceholder.typicode.com

1. Напишите скрипт на Python, который выполняет GET-запрос к этому API и сохраняет первые 10 постов в файл posts.json.
2. Проверьте, чтобы в файле корректно отображались данные в формате JSON.
3. Добавьте обработку ошибок на случай, если сервер недоступен или вернёт ошибку.

Задание 2

Разработка и тестирование функций.

Что нужно сделать:

Создайте модуль Python с функциями для работы с текстом:

1. Функция count_words(text) — должна считать количество слов в строке.
2. Функция find_longest_word(text) — должна находить самое длинное слово в строке.
3. Напишите юнит-тесты для этих функций с использованием библиотеки pytest (не менее 3 тестов на каждую функцию, включая негативные случаи).

Задание 3

Создание простого API на FastAPI.

Что нужно сделать:

1. Создайте небольшое приложение на FastAPI с двумя маршрутами:
    a. GET /ping — возвращает JSON с сообщением { "message": "pong" }.
    b. POST /echo — принимает JSON с полем text и возвращает его же.
2. Запустите приложение локально с помощью Uvicorn.
3. Проверьте работу API с помощью Postman или curl.
4. Реализуйте валидацию входящих данных через Pydantic.

Основная литература

1. Марк Лутц. *Изучаем Python. Том 1. 5-е издание.* — СПб: Вильямс, 2024. — 1216 с. https://www.ozon.ru/product/izuchaem-python-tom-1-5-e-izd-lutts-mark-588939040/

2. Эрик Мэтиз. *Python. Полное руководство для начинающих.* — М.: Эксмо, 2023. — 480 с. https://e.lanbook.com/book/297128

3. В. М. Стасышин, Т. Л. Стасышина. *Базы данных: технологии доступа.* — М.: Юрайт, 2022. — 164 с. https://urait.ru/book/bazy-dannyh-tehnologii-dostupa-494562
4. Официальная документация FastAPI. https://fastapi.tiangolo.com/
5. Максвелл А. *Мощный Python: паттерны и стратегии современного программирования*. — Мягкая обложка. — М.:Питер, 2025. — 208 с.: ил.
6. Жао Э. *SQL. Pocket guide*. – 4-е изд. — Мягкая обложка. — М.:Питер, 2025. — 320 с.: ил.

Дополнительная литература

1. Владимир Соболев. *Чистый Python. Как писать поддерживаемый и читаемый код.* — М.: Манн, Иванов и Фербер, 2021. — 320 с. https://www.mann-ivanov-ferber.ru/books/chistyy-python/
2. Официальная документация Python. https://docs.python.org/3/
3. Официальная документация SQLAlchemy. https://docs.sqlalchemy.org/en/20/
4. Официальная документация Pydantic. https://docs.pydantic.dev/latest/

4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Текущий контроль успеваемости проводится в форме компьютерного тестирования для самоконтроля.

Примерные тестовые вопросы для проведения текущего контроля успеваемости:

1. Какой тип данных будет у значения 5.0 в Python?
- int
- float
- str
- bool

2. Какой метод используется для добавления элемента в список?
- append()
- add()
- insert()
- push()

3. Какой HTTP-метод используется для получения данных с сервера?
- POST
- PUT
- GET
- DELETE

4. Какой декоратор используется в FastAPI для создания маршрута GET?
- @fastapi.get()
- @route.get()
- @app.get()
- @get.app()

5. Какой формат данных чаще всего используется для обмена данными через API?
- XML
- JSON
- CSV
- HTML

6. Какой из перечисленных принципов относится к ООП?
- Валидация
- Инкапсуляция
- Итерирование
- Асинхронность

Итоговая аттестация проходит в форме зачета (компьютерное тестирование)
Тестирование состоит из 30 вопросов закрытого типа.
Общая трудоемкость итоговой аттестации – 4 академических часа.

К итоговой аттестации необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной программе. Систематическое изучение всех лекций, а также самостоятельная работа позволит успешно освоить программу и создать хорошую базу для успешного прохождения итоговой аттестации.

Примерны тестовые вопросы для проведения итоговой аттестации:

1. Как называется структура данных, работающая по принципу "Последним пришёл — первым вышел" (LIFO)?

- Queue (Очередь)
- Stack (Стек)
- List (Список)
- Dictionary (Словарь)

2. Что возвращает функция без оператора return в Python?
- Ноль
- Пустую строку
- None
- Ошибку выполнения

3. Как называется описание структуры данных, передаваемых через API, в FastAPI?
- Schema
- Model
- View
- Template

4. Что создаётся при использовании ключевого слова class в Python?
- Объект
- Экземпляр
- Метод
- Класс

5. Какой ORM чаще всего используется в проектах на Python для работы с реляционными базами данных?
- DjangoORM

- SQLAlchemy
- MongoEngine
- Peewee

6. Что такое Dependency Injection в контексте FastAPI?
- Механизм подключения внешних библиотек
- Передача зависимостей через параметры функций
- Подключение к базе данных
- Внедрение кода в сторонние сервисы

По итогам тестирования слушателю выставляется итоговая оценка.

Критерии оценки обучающихся на итоговой аттестации

Таблица 5

Оценка	Требования к знаниям
зачтено	Выставляется слушателю, если он правильно выполнил не менее 50% заданий
не зачтено	Выставляется слушателю, если он правильно выполнил менее 50% заданий

5. ИНДИКАТОРЫ СФОРМИРОВАННЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКА ПРОГРАММЫ

В результате освоения программы у слушателя сформированы компетенции:

Таблица 6

Профессиональные компетенции или трудовые функции *(формируются и (или) совершенствуются)* ПСК	**Индикаторы освоения компетенции**
ПСК-1. Оформление программного кода в соответствии с установленными требованиями (A/03.3)*	Разработка простых скриптов и утилит. Решение прикладных задач с применением базовых возможностей Python. Обработка данных из файлов.
ПСК-2. Проверка и отладка программного кода (A/05.3)*	Создание и отладка серверных API с использованием FastAPI. Реализация CRUD-операций и взаимодействие с БД. Интеграция внешних API в приложение.
ПСК-3. Разработка тестовых наборов данных для проверки работоспособности компьютерного программного обеспечения (B/02.4)*	Пишет модульные тесты с использованием библиотеки pytest. Проводит отладку кода с применением логирования и IDE-инструментов. Обрабатывает исключения в коде. Применяет архитектурные подходы и проводит рефакторинг проекта. Составляет техническую документацию и отчеты о тестировании.