

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ИННОПРОГ»**

УТВЕРЖДАЮ

Генеральный директор ООО

«ИННОПРОГ»



Венедиктов Р.В.

подпись

/ Венедиктов Р.В.

инициалы, фамилия

01 февраля 2025 г.

Приказ об утверждении
программы

от 01 февраля 2025 г. № ОБР-3

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ**

«Python-разработчик»

Вид программы:	дополнительная профессиональная программа профессиональной переподготовки
Новая квалификация:	Python-разработчик
Трудоемкость:	800 академических часов
Форма обучения:	очная, с применением электронного обучения и дистанционных образовательных технологий
Режим реализации:	10 месяцев (40 учебных недель), рекомендуемая нагрузка – 20 академических часов в неделю
Язык обучения:	русский

Иннополис, 2025

Содержание

1	Общая характеристика программы	5
1.1	Наименование программы	5
1.2	Нормативно-правовые основания разработки программы	5
1.3	Структура образовательной программы	6
1.4	Вид программы и целевая модель результата	7
1.5	Актуальность программы	7
1.6	Цель реализации программы	8
1.7	Задачи программы	8
1.8	Категория слушателей	9
1.9	Требования к уровню подготовки поступающего на обучение . .	9
1.10	Порядок приема, зачисления, отчисления, пересдачи и апелляции	9
1.11	Форма обучения, режим и язык обучения	10
1.12	Срок освоения и трудоемкость программы	11
1.13	Документ о квалификации	11
1.14	Характеристика новой квалификации и связанной с ней профес- сиональной деятельности	11
1.15	Связь Программы с профессиональными стандартами, трудовы- ми функциями и уровнями квалификации	12
1.16	Планируемые результаты освоения программы	14
1.16.1	Перечень формируемых профессиональных компетенций	14
1.16.2	Планируемые результаты обучения	15
2	Содержание программы	18
2.1	Учебный план	18
2.2	Календарный учебный график	21
2.3	Рабочие программы модулей	22
2.3.1	Модуль 1. Введение в профессию и организация обучения	26
2.3.2	Модуль 2. Python начальный	26
2.3.3	Модуль 3. Python продвинутый	30
2.3.4	Модуль 4. ООП в Python	34
2.3.5	Модуль 5. Git	36
2.3.6	Модуль 6. Основы SQL	38
2.3.7	Модуль 7. Алгоритмы и структуры данных	40

2.3.8	Модуль 8. Основы HTML/CSS/JavaScript	43
2.3.9	Модуль 9. Linux	45
2.3.10	Модуль 10. PostgreSQL	47
2.3.11	Модуль 11. Создание Telegram-бота	50
2.3.12	Модуль 12. Django	52
2.4	Проектная практика и консультации	56
3	Организационно-педагогические условия реализации программы	57
3.1	Общие условия реализации	57
3.2	Материально-технические условия	57
3.3	Электронная информационно-образовательная среда	58
3.4	Кадровые условия	59
3.5	Учебно-методическое обеспечение	60
3.6	Применение электронного обучения и дистанционных образо- вательных технологий	60
3.7	Информационная безопасность, персональные данные и этика .	61
3.8	Локальные процедуры обучения	61
3.9	Обновление программы	62
4	Оценка качества освоения программы	63
4.1	Общие положения	63
4.2	Текущий контроль успеваемости	63
4.3	Промежуточная аттестация	64
4.4	Пересдача, апелляция и подтверждение самостоятельности . . .	65
4.5	Допуск к итоговой аттестации	65
4.6	Итоговая аттестация	66
4.6.1	Примерная тематика проектных работ	66
4.6.2	Обязательные требования к итоговому проекту	68
4.6.3	Порядок проведения итоговой аттестации	69
4.6.4	Критерии оценивания итогового проекта	69
4.7	Внутренняя и внешняя оценка качества реализации программы	71
5	Методические материалы	72
5.1	Методические указания для слушателей	72
5.2	Методические указания для преподавателей и проверяющих . .	72

5.3	Рекомендуемая литература и ресурсы	72
	Приложение 1. Матрица формирования компетенций	74
	Приложение 2. Оценочные материалы промежуточной аттестации .	75
	Приложение 3. Оценочные материалы итоговой аттестации	79
	Приложение 4. Перечень локальных нормативных актов, на которые должна опираться реализация программы	83
	Приложение 5. Лист актуализации программы	85
	Приложение 6. Требования к репозиторию, README и демонстрации	86
	Приложение 7. Дорожная карта подготовки выпускного проекта . .	87
	Приложение 8. Контрольный лист соответствия требованиям про- граммы	88

1. Общая характеристика программы

1.1. Наименование программы

Дополнительная профессиональная программа профессиональной переподготовки «Python-разработчик» (далее – Программа) разработана для реализации обществом с ограниченной ответственностью «Иннопрог» в рамках дополнительного профессионального образования.

1.2. Нормативно-правовые основания разработки программы

Программа разработана в соответствии с:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- приказом Министерства науки и высшего образования Российской Федерации от 24.03.2025 № 266 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- постановлением Правительства Российской Федерации от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- постановлением Правительства Российской Федерации от 18.09.2020 № 1490 «О лицензировании образовательной деятельности»;
- постановлением Правительства Российской Федерации от 26.08.2013 № 729 «О федеральной информационной системе «Федеральный реестр сведений о документах об образовании и (или) о квалификации, документах об обучении»»;
- Федеральным законом от 27.07.2006 № 152-ФЗ «О персональных данных»;
- приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н «Об утверждении профессионального стандарта «Программист»»;

- приказом Министерства труда и социальной защиты Российской Федерации от 18.01.2017 № 44н «Об утверждении профессионального стандарта «Разработчик Web и мультимедийных приложений»»;
- ГОСТ Р 7.0.97-2025 «Система стандартов по информации, библиотечно-му и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов»;
- локальными нормативными актами ООО «Иннопрог», регламентирующими разработку и утверждение образовательных программ, организацию образовательного процесса, применение электронного обучения и дистанционных образовательных технологий, текущий контроль, промежуточную и итоговую аттестацию, порядок выдачи документов о квалификации, защиту персональных данных и порядок оказания платных образовательных услуг.

При разработке содержания Программы также учтены требования федеральных государственных образовательных стандартов по укрупненной группе специальностей и направлений подготовки 09.00.00 «Информатика и вычислительная техника» в части результатов, связанных с разработкой программного обеспечения, базами данных, web-технологиями, проектной деятельностью и сопровождением информационных систем.

1.3. Структура образовательной программы

Настоящая Программа «Python-разработчик» оформлена как комплекс основных характеристик образования, организационно-педагогических условий, оценочных и методических материалов, необходимых для реализации дополнительной профессиональной программы профессиональной переподготовки.

В состав Программы включены:

- общая характеристика Программы, включая цель, задачи, категорию слушателей, требования к поступающим, форму обучения, срок освоения, трудоемкость, документ о квалификации и характеристику новой квалификации;

- учебный план, определяющий перечень модулей, трудоемкость, распределение часов по видам учебной работы, формы промежуточной и итоговой аттестации;
- календарный учебный график, определяющий рекомендуемую последовательность освоения модулей и период итоговой аттестации;
- рабочие программы модулей, раскрывающие цели, результаты, тематическое содержание, самостоятельную работу, проектные задания и формы контроля;
- организационно-педагогические условия реализации, включая кадровые, материально-технические, информационно-методические условия, электронную информационно-образовательную среду и порядок применения электронного обучения и дистанционных образовательных технологий;
- оценочные материалы для текущего контроля, промежуточной и итоговой аттестации, критерии допуска, пересдачи, апелляции и подтверждения самостоятельности выполнения работ;
- методические материалы и приложения, обеспечивающие единообразное применение Программы, проверку проектных работ, оформление результатов и актуализацию документа.

Структура Программы обеспечивает проверяемость объема, содержания, планируемых результатов, условий реализации, форм аттестации и документов, выдаваемых по итогам обучения.

1.4. Вид программы и целевая модель результата

Программа относится к дополнительным профессиональным программам профессиональной переподготовки и направлена на получение компетенции, необходимой для выполнения нового вида профессиональной деятельности, а также приобретение новой квалификации в сфере разработки программного обеспечения на языке Python.

1.5. Актуальность программы

Программа обусловлена устойчивым спросом на специалистов, владеющих современным стеком backend-разработки, способных писать код на Python, проектировать и использовать базы данных, работать с системой контроля вер-

сий, разрабатывать Telegram-ботов и веб-приложения на Django, а также сопровождать и разворачивать программные продукты в Linux-среде. Программа ориентирована на практико-ориентированное формирование компетенций, востребованных в коммерческой разработке, в продуктовых и сервисных командах, а также в сфере цифровизации внутренних процессов организаций.

1.6. Цель реализации программы

Цель реализации Программы – формирование у слушателя комплекса профессиональных компетенций, необходимых для выполнения нового вида профессиональной деятельности в сфере backend-разработки на языке Python, проектирования и разработки прикладных программ, web-приложений, Telegram-ботов и сервисов, взаимодействующих с реляционными базами данных и web-интерфейсами.

1.7. Задачи программы

Задачами Программы являются:

- формирование устойчивой базы по синтаксису и идиомам Python;
- развитие алгоритмического мышления и навыков решения прикладных задач;
- освоение принципов объектно-ориентированного проектирования;
- формирование навыков командной разработки с использованием Git;
- освоение реляционных баз данных, языка SQL и СУБД PostgreSQL;
- приобретение навыков создания Telegram-ботов и web-приложений на Django;
- освоение базовой разработки API, интеграции с внешними сервисами и безопасной работы с пользовательскими данными;
- формирование навыков тестирования через `pytest`, логирования, работы с переменными окружения и конфигурациями;
- формирование навыков подготовки программного продукта к разворачиванию и сопровождению с использованием Linux, Docker, серверов приложений и базовых CI/CD-практик;

- развитие культуры инженерной разработки: документирование, тестирование, читаемость, структурированность и сопровождение кода.

1.8. Категория слушателей

К освоению Программы допускаются:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

1.9. Требования к уровню подготовки поступающего на обучение

Обязательным условием допуска к освоению Программы является наличие образования или получение образования, соответствующего требованиям законодательства к дополнительным профессиональным программам.

Рекомендуемый входной уровень подготовки:

- уверенное владение персональным компьютером и современными интернет-сервисами;
- базовая цифровая грамотность;
- готовность к самостоятельной работе и регулярному выполнению практических заданий;
- мотивация к освоению новой квалификации в сфере разработки программного обеспечения.

1.10. Порядок приема, зачисления, отчисления, пересдачи и апелляции

Прием на обучение осуществляется на основании заявления слушателя, документа, удостоверяющего личность, документа о среднем профессиональном и (или) высшем образовании либо справки об обучении для лица, получающего соответствующее образование, а также договора об образовании на обучение по дополнительной профессиональной программе. Перечень документов, порядок их представления, сроки рассмотрения заявления и основания отказа в приеме определяются локальными нормативными актами организации и договором об образовании.

Зачисление слушателя оформляется приказом организации после заключения договора, проверки входных документов и предоставления доступа к электронной информационно-образовательной среде. Отчисление, восстановление, перевод на индивидуальный график, изменение сроков обучения и прекращение образовательных отношений осуществляются в порядке, установленном локальными нормативными актами организации и договором об образовании.

Порядок пересдачи текущего контроля, промежуточной аттестации и итоговой аттестации, сроки устранения замечаний, основания допуска к повторной сдаче и порядок подачи апелляции определяются локальным актом о текущем контроле, промежуточной и итоговой аттестации. Слушателю обеспечивается возможность получить мотивированную обратную связь по результатам проверки и представить доработанную работу в установленные сроки.

1.11. Форма обучения, режим и язык обучения

Обучение по Программе осуществляется в очной форме с применением электронного обучения и дистанционных образовательных технологий. Электронное обучение и дистанционные образовательные технологии применяются при проведении учебных занятий, консультаций, самостоятельной работы, текущего контроля, промежуточной аттестации, проектной практики и итоговой аттестации при условии обеспечения доступа слушателя к электронной информационно-образовательной среде, учебным материалам, заданиям, средствам коммуникации и процедурам идентификации слушателя.

Язык обучения – русский. Для всех видов учебных занятий академический час устанавливается продолжительностью 45 минут. Образовательный процесс может осуществляться в течение всего календарного года. Конкретные даты начала и окончания обучения определяются приказом о зачислении, договором об образовании, расписанием и календарным учебным графиком.

Рекомендуемый режим освоения Программы – 20 академических часов в неделю в течение 10 месяцев (40 учебных недель). Допускается обучение по индивидуальному учебному плану в порядке, установленном локальным нормативным актом организации, при условии сохранения общей трудоемкости Программы и достижения планируемых результатов.

1.12. Срок освоения и трудоемкость программы

Трудоемкость Программы составляет 800 академических часов, включая все виды учебной работы слушателя, текущий контроль, промежуточную аттестацию и итоговую аттестацию.

Рекомендуемый срок освоения Программы составляет 10 месяцев (40 учебных недель). Допускается реализация Программы в ином календарном периоде при условии сохранения ее общей трудоемкости и достижения заявленных результатов обучения.

1.13. Документ о квалификации

Лицам, успешно освоившим Программу и прошедшим итоговую аттестацию, выдается диплом о профессиональной переподготовке установленного организацией образца.

Лицам, получающим среднее профессиональное и (или) высшее образование, диплом о профессиональной переподготовке выдается одновременно с получением соответствующего документа об образовании и о квалификации.

Лицам, не прошедшим итоговую аттестацию или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть Программы и (или) отчисленным из организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно установленному организацией.

Сведения о выданных дипломах о профессиональной переподготовке подлежат внесению в федеральную информационную систему «Федеральный реестр сведений о документах об образовании и (или) о квалификации, документах об обучении» (ФИС ФРДО) в случаях, составе и порядке, установленных законодательством Российской Федерации и локальным актом организации о выдаче и учете документов о квалификации.

1.14. Характеристика новой квалификации и связанной с ней профессиональной деятельности

Новая квалификация, приобретаемая по итогам освоения Программы: **Python-разработчик.**

Область профессиональной деятельности выпускника:

- разработка, модификация, тестирование и сопровождение программного обеспечения;
- разработка серверной логики web-приложений;
- создание Telegram-ботов, API и сервисов обработки данных;
- проектирование, использование и сопровождение реляционных баз данных;
- участие в развертывании и техническом сопровождении приложений.

Объектами профессиональной деятельности выпускника являются:

- программный код и программные модули;
- базы данных и схемы хранения данных;
- web-приложения, Telegram-боты, API и вспомогательные сервисы;
- репозитории исходного кода, техническая документация, средства автоматизации и развертывания;
- пользовательские интерфейсы и серверные компоненты информационных систем.

Типовые виды профессиональной деятельности выпускника:

- разработка прикладных программ на Python;
- разработка backend-компонентов web-приложений;
- разработка Telegram-ботов и интеграционных сервисов;
- проектирование и использование баз данных;
- поддержка и развитие существующих программных решений;
- участие в командной разработке и выпуске версий программного продукта.

1.15. Связь Программы с профессиональными стандартами, трудовыми функциями и уровнями квалификации

Содержание Программы ориентировано на формирование компетенций для нового вида профессиональной деятельности в сфере разработки программного обеспечения. Основным профессиональным стандартом для соотнесения результатов обучения является профессиональный стандарт «Программист». Профессиональный стандарт «Разработчик Web и мультимедий-

ных приложений» учитывается в части web-интерфейсов, шаблонов, клиент-серверного взаимодействия и сопровождения web-ресурсов.

Таблица 1 – Связь программы с профессиональными стандартами и трудовыми функциями

Профессиональный стандарт / блок	Трудовые функции, учитываемые при разработке Программы	Уровень квалификации	Модули и компетенции Программы
ПС «Программист», ОТФ А «Разработка и отладка программного кода»	А/01.3 – формализация и алгоритмизация задач; А/02.3 – написание программного кода; А/03.3 – оформление программного кода; А/04.3 – работа с системой управления версиями; А/05.3 – проверка и отладка кода.	3	Модули 2–7, 13–14; ПК-1, ПК-2, ПК-3, ПК-5.
ПС «Программист», ОТФ В «Проверка работоспособности и рефакторинг кода программного обеспечения»	В/01.4 – разработка процедур проверки; В/02.4 – разработка тестовых наборов данных; В/03.4 – проверка работоспособности; В/04.4 – рефакторинг, оптимизация и инспекция кода; В/06.4 – сборка программных модулей в проект.	4	Модули 3–7, 10, 12–14; ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-8, ПК-10.
ПС «Программист», ОТФ С «Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта»	С/01.5 – разработка процедур интеграции; С/02.5 – интеграция модулей и проверка работоспособности выпусков.	5	Модули 9–14; ПК-3, ПК-4, ПК-7, ПК-8, ПК-9, ПК-10.

Профессиональный стандарт / блок	Трудовые функции, учитываемые при разработке Программы	Уровень квалификации	Модули и компетенции Программы
ПС «Разработчик Web и мультимедийных приложений»	Разработка и сопровождение web-страниц, web-интерфейсов, интерактивных компонентов, работа с клиентской частью и интеграция с серверной логикой.	4–5	Модули 8, 11, 12, 13–14; ПК-6, ПК-9, ПК-10.

Программа не заявляет подготовку к выполнению всех трудовых функций профессиональных стандартов в полном объеме. Перечень трудовых функций используется для определения целевых результатов обучения, уровня сложности заданий, требований к итоговому проекту и критериев оценивания.

1.16. Планируемые результаты освоения программы

1.16.1. Перечень формируемых профессиональных компетенций

Таблица 2 – Формируемые профессиональные компетенции

Код	Наименование профессиональной компетенции
ПК-1	Разрабатывать консольные и серверные программы на языке Python, применять базовые и продвинутое конструкции языка.
ПК-2	Проектировать программные компоненты с использованием объектно-ориентированного подхода и принципов качественного кода.
ПК-3	Использовать Git для индивидуальной и командной разработки, сопровождения и выпуска версий.
ПК-4	Проектировать реляционные структуры данных и писать SQL-запросы различной сложности.

Код	Наименование профессиональной компетенции
ПК-5	Выбирать и реализовывать алгоритмы и структуры данных с учетом требований задачи.
ПК-6	Понимать основы клиентской части веб-приложений, работать с HTML, CSS и JavaScript на уровне взаимодействия backend/frontend.
ПК-7	Использовать Linux-среду и базовые административные инструменты для запуска и сопровождения приложений.
ПК-8	Использовать PostgreSQL, транзакции, индексы, ORM и средства интеграции БД с Python-приложением.
ПК-9	Разрабатывать прикладные Telegram-боты на Python, проектировать диалоги и интегрировать внешние сервисы.
ПК-10	Разрабатывать веб-приложения на Django, включая модели, шаблоны, авторизацию, API, middleware и деплой.

1.16.2. Планируемые результаты обучения

В результате освоения Программы слушатель должен:

знать:

- синтаксис и основные конструкции языка Python;
- принципы алгоритмизации, структуры данных и типовые алгоритмы;
- основы объектно-ориентированного проектирования;
- основы командной разработки и контроля версий;
- основы проектирования реляционных баз данных, SQL и PostgreSQL;
- базовые принципы построения клиентской части web-приложений;
- устройство и базовые средства администрирования Linux-среды;
- архитектуру Telegram-ботов и web-приложений на Django;
- основы web-взаимодействия, HTTP, клиент-серверной модели и API;
- требования к качеству кода, базовые подходы к тестированию и подготовке проекта к развертыванию;

- назначение Docker, переменных окружения, серверов приложений, reverse проху и базовых CI/CD-процессов;
- базовые требования к безопасности web-приложений: хранение секретов, разграничение доступа, защита форм и проверка пользовательского ввода.

уметь:

- писать, читать и отлаживать код на Python;
- проектировать и использовать классы, функции, модули и пакеты;
- работать с файлами, исключениями и внешними библиотеками;
- использовать Git для сопровождения проекта и командной работы;
- писать SQL-запросы и проектировать структуру реляционной базы данных;
- создавать серверную логику и web-приложения на Django;
- разрабатывать Telegram-ботов и интегрировать их с базой данных и внешними сервисами;
- использовать Linux-командную строку и базовые средства настройки окружения;
- писать базовые тесты, настраивать логирование и хранить конфигурацию проекта через переменные окружения;
- подготавливать проект к запуску через Docker, сервер приложений и базовые сценарии CI/CD;
- документировать решения, подготавливать проект к передаче и защите.

владеть навыками:

- практической разработки программ на Python;
- выбора и применения структур данных и алгоритмов;
- организации проекта и работы с исходным кодом в репозитории;
- проектирования и сопровождения БД;
- разработки backend-логики web-приложения и Telegram-бота;
- взаимодействия с ORM и SQL-инструментами;
- запуска и сопровождения проекта в Linux-среде;
- базового контейнеризированного запуска, настройки переменных окружения, ведения логов и подготовки инструкции для деплоя;

- выполнения итогового проекта от постановки задачи до защиты результата.

2. Содержание программы

2.1. Учебный план

Таблица 3 – Учебный план

№	Наименование модуля	Всего, час.	Лекц., час.	Практ., час.	Сам. раб., час.	ПА, час.	ИА, час.	Форма контроля
1	Введение в профессию и организация обучения	12	6	3	3	0	0	входной контроль
2	Python начальный	116	41	46	23	6	0	зачет
3	Python продвинутый	80	26	31	17	6	0	зачет
4	ООП в Python	56	20	20	11	5	0	зачет
5	Git	36	12	12	6	6	0	зачет
6	Основы SQL	36	12	12	6	6	0	зачет
7	Алгоритмы и структуры данных	68	23	23	17	5	0	зачет
8	Основы HTML/CSS/JavaScript	56	17	20	14	5	0	зачет
9	Linux	44	14	14	11	5	0	зачет
10	PostgreSQL	68	23	23	17	5	0	зачет
11	Создание Telegram-бота	56	17	20	14	5	0	зачет
12	Django	104	32	37	26	9	0	зачет
13	Проектная практика и консультации	36	6	18	12	0	0	допуск к ИА
14	Итоговая аттестация	32	0	0	0	0	32	защита проекта
Итого		800	249	279	177	63	32	–



Промежуточная аттестация осуществляется по завершении модулей, определенных учебным планом. Итоговая аттестация проводится отдельно от промежуточной аттестации в форме защиты итогового аттестационного проекта. Трудоемкость итоговой аттестации составляет 32 академических часа.

2.2. Календарный учебный график

Начало обучения определяется по мере комплектования учебной группы. Учебные группы могут формироваться в течение всего календарного года. Период освоения Программы определяется календарным учебным графиком и расписанием. Ниже представлен рекомендуемый график при сроке обучения 10 месяцев (40 недель).

Таблица 4 – Календарный учебный график

№	Наименование модуля	Период обучения	Часы
1	Введение в профессию и организация обучения	1 неделя	12
2	Python начальный	2-7 недели	116
3	Python продвинутый	8-11 недели	80
4	ООП в Python	12-14 недели	56
5	Git	15-16 недели	36
6	Основы SQL	17-18 недели	36
7	Алгоритмы и структуры данных	19-21 недели	68
8	Основы HTML/CSS/JavaScript	22-24 недели	56
9	Linux	25-26 недели	44

№	Наименование модуля	Период обучения	Часы
10	PostgreSQL	27-29 недели	68
11	Создание Telegram-бота	30-32 недели	56
12	Django	33-37 недели	104
13	Проектная практика и консультации	38-39 недели	36
14	Итоговая аттестация	40 неделя	32
Итого			800

2.3. Рабочие программы модулей

В рабочих программах модулей определяются цели, результаты, тематическое содержание, формы организации занятий, самостоятельной работы и контроля. Перечень тем соотносится с фактической структурой учебных материалов и внутреннего контента направления «Python-разработчик».

Проектная траектория программы. Практическая часть Программы организована через последовательную разработку проектных работ. Проекты используются в текущем контроле, промежуточной аттестации, проектной практике и при подготовке итогового аттестационного проекта. Сложность работ повышается от консольных программ к Telegram-ботам, web-приложениям, базам данных, Django-проектам и развертыванию приложения в Linux-среде. Перечень проектных работ приведен ниже.

Таблица 5 – Проектная траектория программы

№	Проект	Содержание проектной работы	Связанные модули
1	Система учета фруктов на складе	Консольная программа для учета товаров на складе: добавление позиций, контроль остатков, обновление количества и формирование понятного отчета по доступной продукции.	Python начальный
2	База зарплат сотрудников	Система для хранения данных о сотрудниках, управления зарплатами и быстрого поиска информации, направленная на автоматизацию базовых HR- и бухгалтерских операций.	Python начальный
3	Менеджер задач в консоли	Приложение для постановки, отслеживания и закрытия задач, позволяющее управлять личными или рабочими делами в одном месте.	Python начальный / Python продвинутый
4	Telegram-бот для управления задачами	Telegram-бот для добавления задач, просмотра списка дел и отметки выполненных пунктов прямо в мессенджере.	Создание Telegram-бота
5	Сервис бронирования поездок	Система для оформления поездок, учета клиентов и транспорта, моделирующая работу небольшого сервиса заказа машин и автоматизацию бронирования.	Python продвинутый / ООП в Python

№	Проект	Содержание проектной работы	Связанные модули
6	Сервис управления пассажирскими перевозками	Приложение для учета транспорта, вместимости и пассажиров, моделирования посадки, контроля загрузки и отслеживания доступных мест.	ООП в Python / SQL / PostgreSQL
7	Складской менеджер товаров	Система управления ассортиментом магазина с учетом остатков, цен и доступности товаров, поддерживающая быстрый поиск позиций и контроль склада.	Python продвинутый / SQL / PostgreSQL
8	Key-Value хранилище на дереве поиска	Собственная система хранения данных с добавлением, поиском, обновлением и удалением записей на основе дерева поиска.	Алгоритмы и структуры данных
9	Веб-приложение прогноза погоды	Современный сайт для отображения актуальной погоды по выбранному городу с использованием внешнего API, удобного интерфейса и понятной визуализации данных.	HTML/CSS/JavaScript
10	Telegram-бот с прогнозом погоды	Бот, который по запросу пользователя показывает прогноз погоды и демонстрирует работу с внешним API и Telegram-интерфейсом.	Создание Telegram-бота

№	Проект	Содержание проектной работы	Связанные модули
11	Telegram-бот для бронирования столиков	Бот для ресторанов и кафе, через который клиенты выбирают дату, время и количество гостей, а бизнес принимает и обрабатывает заявки.	Создание Telegram-бота
12	Telegram-магазин с каталогом и корзиной	Telegram-бот с каталогом товаров, карточками, корзиной и логикой оформления заказа для продаж в мессенджере.	Создание Telegram-бота / PostgreSQL
13	Django-приложение интернет-магазина	Интернет-магазин на Django с каталогом товаров, карточками, изображениями и структурированной базой данных для управления ассортиментом.	Django
14	Интернет-магазин с регистрацией и корзиной	Е-commerce проект с авторизацией пользователей, корзиной, добавлением и удалением товаров, приближенный к коммерческой разработке.	Django
15	Продакшн-версия интернет-магазина на Django	Веб-проект на Django с PostgreSQL, административной панелью, серверной настройкой, Docker-запуском, базовым CI/CD, деплоем на Linux и архитектурой, приближенной к требованиям бизнеса.	Django / Linux / PostgreSQL

Слушатель может выполнять указанные проекты последовательно либо выбрать индивидуальную тему после согласования с преподавателем. Инди-

видуальная тема должна соответствовать результатам Программы, включать Python-разработку, работу с данными, понятный пользовательский сценарий, репозиторий и материалы для проверки.

2.3.1. Модуль 1. Введение в профессию и организация обучения

Цель модуля – ввести слушателя в программу профессиональной переподготовки, определить требования к итоговому проекту, критерии качества учебной работы и организацию обучения в дистанционной среде.

Объем модуля: 12 часов.

Краткое содержание:

- область деятельности Python-разработчика и место профессии в ИТ-проектах;
- обзор стека программы: Python, Git, SQL, PostgreSQL, Linux, Telegram, Django;
- правила работы в электронной информационно-образовательной среде;
- порядок текущего контроля, промежуточной аттестации и итоговой аттестации;
- требования к репозиторию, структуре проекта, коммуникации и срокам сдачи работ.

Самостоятельная работа: настройка учетных записей, рабочего окружения, среды разработки и репозитория для последующего выполнения практических заданий.

Форма контроля: входной организационный контроль без выставления оценки.

Проектная ориентация модуля: слушатель знакомится с перечнем проектных работ Программы, требованиями к репозиторию, описанию результата, срокам сдачи и критериям защиты итогового проекта.

2.3.2. Модуль 2. Python начальный

Цель модуля: сформировать устойчивые навыки алгоритмизации и программирования на языке Python на уровне базовых конструкций языка, функций и стандартных модулей.

Объем модуля: 116 часов.

Планируемые результаты по модулю:

- создавать и отлаживать консольные программы на Python;
- использовать переменные, базовые типы данных, арифметические и логические операции;
- применять условные конструкции, циклы, функции и стандартные модули;
- решать типовые задачи обработки строк, списков, словарей и кортежей.

Тематический план модуля:

Таблица 6 – Тематический план модуля 2 «Python начальный»

№	Тема	Часы	Краткое содержание
1	Логические операции	9	булевы значения, операции сравнения, составные условия, приоритет логических выражений.
2	IDE	9	установка и настройка среды разработки, запуск программ, навигация по проекту, базовая отладка.
3	Переменные в Python	8	переменные, присваивание, именованное, чтение и изменение значений в памяти программы.
4	Типы данных	8	числа, строки, булев тип, преобразование типов, проверка типа и ограничения операций.
5	Вывод данных, f-строки	8	функция print, форматированный вывод, f-строки, экранирование и читаемость результата.

№	Тема	Часы	Краткое содержание
6	Арифметические операции	8	операции над числами, приоритет выражений, целочисленное деление, остаток от деления.
7	Ввод данных	6	функция input, преобразование пользовательского ввода, валидация простых данных.
8	Условный оператор	6	конструкции if/elif/else, вложенные ветвления, типовые сценарии принятия решений.
9	Отладка программы	6	поиск и исправление ошибок, чтение traceback, тестовые примеры, пошаговая отладка.
10	Тернарный оператор	6	краткая запись ветвления, уместное использование и требования к читаемости.
11	Строки	6	индексация, срезы, методы строк, поиск и замена, базовая обработка текстовых данных.
12	Списки	6	создание списков, индексация, изменение элементов, методы и базовые алгоритмы обработки коллекций.
13	Цикл while	6	циклы с предусловием, условие завершения, защита от бесконечных циклов.

№	Тема	Часы	Краткое содержание
14	Цикл for	6	итерация по последовательностям, диапазоны, вложенные циклы, шаблоны перебора данных.
15	Словари, кортежи	6	ассоциативные массивы, неизменяемые структуры, доступ к данным и практические сценарии.
16	Функции	6	объявление и вызов функций, параметры, возвращаемые значения, декомпозиция программы.
17	Модуль math, random	6	работа со стандартными модулями, импорт, генерация случайных значений и математические функции.

Содержание модуля: В модуле рассматриваются синтаксис Python, работа в среде разработки, базовые типы данных, правила ввода и вывода, форматирование строк, арифметические и логические операции, ветвления, отладка и базовые структуры данных. Существенное внимание уделяется циклам, функциям, параметрам, возвращаемым значениям, а также практическому применению модулей math и random. Практическая часть ориентирована на решение учебных задач, создание мини-программ, написание функций и постепенную подготовку слушателя к следующему уровню обучения.

Практические виды работ:

- написание консольных программ с пользовательским вводом;
- разработка набора задач на условия и циклы;
- обработка строк, списков и словарей;

- создание библиотеки простых пользовательских функций.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: тестирование на платформе, проверка практических заданий, промежуточный зачет в форме решения набора задач.

Проектные задания модуля:

- **Система учета фруктов на складе** – разработка консольной программы для добавления товаров, контроля остатков, изменения количества и формирования отчета по доступной продукции.
- **База зарплат сотрудников** – создание программы для хранения данных о сотрудниках, управления зарплатами и быстрого поиска информации.
- **Менеджер задач в консоли** – разработка приложения для постановки, отслеживания и закрытия задач с использованием базовых структур данных Python.

2.3.3. Модуль 3. Python продвинутый

Цель модуля: сформировать у слушателей навыки разработки структурированных и сопровождаемых Python-приложений с использованием модулей, пакетов, файлового ввода-вывода, исключений, итераторов, генераторов, декораторов, контекстных менеджеров, виртуальных окружений и базового тестирования.

Объем модуля: 80 часов.

Планируемые результаты по модулю:

- организовывать код в виде модулей и пакетов, использовать виртуальные окружения и зависимости;
- работать с файлами, каталогами, форматами CSV/JSON и исключительными ситуациями;
- применять итераторы, генераторы, списковые включения, функции высшего порядка и декораторы;

- использовать аннотации типов, `dataclasses`, контекстные менеджеры и базовые модульные тесты.

Тематический план модуля:

Таблица 7 – Тематический план модуля 3 «Python продвинутый»

№	Тема	Часы	Краткое содержание
1	Области видимости и пространство имен	6	правила LEGB, локальная и глобальная области видимости, замыкания, типовые ошибки проектирования функций.
2	Модули, пакеты и структура проекта	6	импорт, абсолютные и относительные импорты, файл <code>__init__.py</code> , повторное использование кода.
3	Работа с файлами и форматами данных	8	<code>pathlib</code> , чтение и запись файлов, кодировки, CSV, JSON, безопасная работа с путями.
4	Исключения, логирование и надежность кода	8	<code>try/except/else/finally</code> , пользовательские исключения, логирование, обработка ошибок и восстановление.
5	Итераторы и генераторы	8	протокол итерации, <code>yield</code> , ленивые вычисления, обработка последовательностей данных.
6	Списковые, словарные и множественные включения	6	преобразование коллекций, фильтрация, читаемость и ограничения использования включений.

№	Тема	Часы	Краткое содержание
7	Функции как объекты и функции высшего порядка	6	передача функций как аргументов, <code>lambda</code> , <code>map/filter/sorted</code> , проектирование чистых функций.
8	Декораторы и замыкания	8	декораторы без аргументов и с аргументами, <code>functools.wraps</code> , практические сценарии применения.
9	Контекстные менеджеры	6	оператор <code>with</code> , протокол контекстного менеджера, <code>contextlib</code> , управление ресурсами.
10	Виртуальные окружения, зависимости и конфигурация	6	<code>venv</code> , <code>pip</code> , файл зависимостей, переменные окружения и воспроизводимый запуск проекта.
11	Аннотации типов и <code>dataclasses</code>	6	базовые типы, коллекции, <code>Optional</code> , <code>dataclass</code> , улучшение читаемости и поддержки кода.
12	Основы тестирования Python-кода	6	<code>pytest/unittest</code> , тестовые сценарии, фиксация дефектов и регрессионная проверка.

Содержание модуля: Модуль расширяет базовую подготовку и переводит слушателя от написания отдельных скриптов к разработке небольших поддерживаемых приложений. Особое внимание уделяется воспроизводимой структуре проекта, работе с данными, обработке исключений, тестированию,

логированию, конфигурации через переменные окружения и оформлению результата в репозитории.

Практические виды работ:

- разработка файловой утилиты для чтения, преобразования и сохранения данных CSV/JSON;
- декомпозиция программы на модули и пакеты с настройкой виртуального окружения;
- реализация генераторов, декораторов и контекстных менеджеров для прикладных задач;
- написание базовых тестов, добавление логирования и исправление дефектов по результатам проверки.

Самостоятельная работа: изучение материалов, выполнение заданий на образовательной платформе, оформление кода в Git-репозитории, анализ замечаний преподавателя, доработка решений и повторная сдача при необходимости.

Форма текущего и промежуточного контроля: контрольные задания на платформе, практикум по обработке файлов и модульной структуре проекта, промежуточный зачет.

Проектные задания модуля:

- доработка проекта **Менеджер задач в консоли**: разбиение кода на функции и модули, обработка ошибок, сохранение данных и подготовка инструкций по запуску;
- **Сервис бронирования поездов** – реализация логики оформления поездов, учета клиентов и транспорта;
- **Складской менеджер товаров** – развитие складской логики с учетом остатков, цен, доступности товаров и поиска позиций.

2.3.4. Модуль 4. ООП в Python

Цель модуля: сформировать у слушателя навыки проектирования программ в объектно-ориентированной парадигме на языке Python.

Объем модуля: 56 часов.

Планируемые результаты по модулю:

- создавать классы и объекты, проектировать интерфейсы классов;
- использовать наследование, инкапсуляцию и полиморфизм;
- применять магические методы, свойства, итераторы и декораторы;
- анализировать и улучшать архитектуру программ с позиции ООП.

Тематический план модуля:

Таблица 8 – Тематический план модуля 4 «ООП в Python»

№	Тема	Часы	Краткое содержание
1	Введение	8	обзор предметной области, критерии качества, место модуля в общей траектории.
2	Классы и объекты	8	создание классов, экземпляров, конструкторов и методов экземпляра.
3	Self и атрибуты класса	6	роль self, атрибуты экземпляра и класса, правила доступа к состоянию.
4	Декораторы в ООП	6	свойства, статические и классовые методы, прикладные сценарии использования.
5	Наследование	6	расширение поведения базовых классов, переопределение методов, композиция и наследование.

№	Тема	Часы	Краткое содержание
6	Важные принципы ООП	6	инкапсуляция, полиморфизм, абстракция, базовые принципы SOLID.
7	Магические методы	6	перегрузка поведения объектов, сравнение, строковое представление, протоколы.
8	Итераторы	5	итераторы и генераторы в контексте ООП, пользовательские последовательности.
9	Заключение	5	систематизация пройденного материала и защита практической работы.

Содержание модуля: Модуль посвящен объектно-ориентированному программированию: построению классов, работе с экземплярами и атрибутами, декораторам в ООП, принципам наследования, композиции и основам SOLID. Рассматриваются магические методы, протоколы итерации и способы повышения выразительности объектной модели. Практика строится на проектировании доменных сущностей, написании пользовательских контейнеров и постепенном переходе от процедурного стиля к объектно-ориентированному.

Практические виды работ:

- реализация предметной модели в виде набора связанных классов;
- написание классов с магическими методами и свойствами;
- создание итераторов и пользовательских коллекций;
- рефакторинг процедурного кода в объектную модель.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: практическая работа по проектированию классов и итоговый зачет по модулю.

Проектные задания модуля:

- **Сервис бронирования поездок** – выделение сущностей клиента, транспорта, поездки и бронирования, описание методов и ограничений;
- **Сервис управления пассажирскими перевозками** – проектирование классов транспорта, пассажира и рейса, контроль вместимости и доступных мест;
- **Складской менеджер товаров** – построение объектной модели товара, склада, операций поступления и списания.

2.3.5. Модуль 5. Git

Цель модуля: сформировать навыки использования системы контроля версий Git в индивидуальной и командной разработке.

Объем модуля: 36 часов.

Планируемые результаты по модулю:

- создавать локальные и удаленные репозитории;
- фиксировать изменения, работать с ветками и историей коммитов;
- выполнять слияние и разрешать конфликты;
- организовывать совместную разработку через удаленные хранилища.

Тематический план модуля:

Таблица 9 – Тематический план модуля 5 «Git»

№	Тема	Часы	Краткое содержание
1	Начало работы	9	создание репозитория, настройка пользователя, базовые команды и жизненный цикл коммита.
2	Изменения	9	индекс, статус файлов, коммиты, откат локальных изменений, просмотр истории.

№	Тема	Часы	Краткое содержание
3	Удаленное хранилище	6	работа с удаленным репозиторием, clone, push, pull, базовые правила совместной работы.
4	Ветки изменений	6	ветвление, merge, rebase на базовом уровне, разрешение конфликтов.
5	Заключение	6	систематизация пройденного материала и защита практической работы.

Содержание модуля: В модуле изучаются основы Git: создание репозитория, индексация и фиксация изменений, работа с историей и ветвлением, взаимодействие с удаленными репозиториями и базовые правила командной разработки. Практический акцент делается на осмысленное ведение истории проекта, оформлении коммитов, pull/push-сценариях и использовании Git как обязательной части итогового проекта программы.

Практические виды работ:

- инициализация репозитория и ведение истории коммитов;
- работа с ветками feature/fix и слияние изменений;
- размещение проекта в удаленном репозитории;
- разрешение типовых merge-conflict ситуаций.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: выполнение практического кейса в Git и зачет по журналу коммитов и структуре репозитория.

Проектные задания модуля: слушатель оформляет один из ранее выполненных проектов в Git-репозитории, ведет историю изменений, фиксирует

этапы разработки, подготавливает README, структуру веток и описание порядка запуска.

2.3.6. Модуль 6. Основы SQL

Цель модуля: сформировать базовые навыки проектирования простых реляционных структур данных и написания SQL-запросов для хранения, отбора, изменения и анализа данных.

Объем модуля: 36 часа.

Планируемые результаты по модулю:

- объяснять назначение реляционной модели данных, таблиц, ключей и связей;
- создавать простые таблицы и выполнять операции добавления, изменения и удаления данных;
- писать запросы **SELECT** с фильтрацией, сортировкой, ограничением и агрегацией;
- применять соединения таблиц и подзапросы в типовых задачах backend-разработки.

Тематический план модуля:

Таблица 10 – Тематический план модуля 6 «Основы SQL»

№	Тема	Часы	Краткое содержание
1	Реляционная модель данных и SQLite	6	таблицы, строки, столбцы, первичные и внешние ключи, типы данных, назначение SQLite в учебном практикуме.
2	DDL: создание и изменение структуры данных	6	CREATE TABLE , ограничения целостности, базовая нормализация, изменение и удаление таблиц.

№	Тема	Часы	Краткое содержание
3	Базовые запросы SELECT	6	SELECT, WHERE, ORDER BY, LIMIT, операторы сравнения, логические условия.
4	Изменение данных	6	INSERT, UPDATE, DELETE, безопасное изменение записей, проверка результата операций.
5	Агрегация и группировка	6	COUNT, SUM, AVG, MIN, MAX, GROUP BY, HAVING.
6	JOIN, подзапросы и итоговый практикум	6	INNER/LEFT JOIN, связи между таблицами, подзапросы, комплексная задача по учебной базе данных.

Содержание модуля: В модуле изучаются основы SQL на примере учебной базы данных. Слушатели последовательно проходят путь от проектирования простых таблиц до написания запросов с фильтрацией, агрегацией и соединением таблиц. Практическая часть ориентирована на ручное написание SQL-команд, понимание результата запроса и подготовку к дальнейшей работе с PostgreSQL и ORM.

Практические виды работ:

- создание учебной базы данных и таблиц с ключами;
- наполнение базы тестовыми данными;
- написание SQL-запросов на выборку, фильтрацию, агрегацию и соединение таблиц;
- выполнение комплексного практикума с проверкой корректности результата.

Самостоятельная работа: выполнение набора SQL-задач на образовательной платформе, оформление запросов и результатов, исправление ошибок по замечаниям преподавателя.

Форма текущего и промежуточного контроля: тестирование по основам реляционной модели и практический зачет по написанию SQL-запросов.

Проектные задания модуля:

- проектирование таблиц для проектов **База зарплат сотрудников, Сервис бронирования поездок, Сервис управления пассажирскими перевозками и Складской менеджер товаров;**
- написание SQL-запросов для добавления, обновления, выборки, фильтрации и агрегации данных в выбранной предметной области.

2.3.7. Модуль 7. Алгоритмы и структуры данных

Цель модуля: развить алгоритмическое мышление и сформировать навыки выбора структур данных и алгоритмов для решения прикладных задач backend-разработки.

Объем модуля: 68 часов.

Планируемые результаты по модулю:

- оценивать временную и пространственную сложность алгоритмов в нотации Big O;
- применять поиск, сортировку, рекурсию и динамическое программирование;
- использовать массивы, списки, стек, очередь, словари, деревья и графы;
- аргументировать выбор алгоритма и структуры данных под конкретную задачу.

Тематический план модуля:

Таблица 11 – Тематический план модуля 7 «Алгоритмы и структуры данных»

№	Тема	Часы	Краткое содержание
1	Введение в анализ алгоритмов	6	понятие алгоритма, корректность, временная и пространственная сложность, Big O, сравнение решений.

№	Тема	Часы	Краткое содержание
2	Линейный и бинарный поиск	6	последовательный поиск, бинарный поиск, требования к данным, типовые ошибки границ и индексов.
3	Сортировки простого уровня	6	сортировка выбором, вставками, пузырьковая сортировка, устойчивость, сравнение сложности.
4	Рекурсия	6	базовый и рекурсивный случаи, стек вызовов, рекурсивный обход данных, ограничения рекурсии.
5	Быстрая сортировка и сортировка слиянием	6	разделяй и властвуй, опорный элемент, слияние, сравнение практической эффективности.
6	Абстрактные структуры данных	6	интерфейс структуры данных, массив, динамический список, связный список, операции и ограничения.
7	Стек и очередь	6	LIFO/FIFO, очередь задач, стек вызовов, проверка скобочных последовательностей, практические сценарии.
8	Хеш-таблицы и словари	6	хеширование, коллизии, словарь Python, множества, ускорение поиска и учета данных.

№	Тема	Часы	Краткое содержание
9	Деревья	5	бинарное дерево, дерево поиска, обходы в глубину и ширину, применение в индексах и иерархиях.
10	Графы	5	вершины и ребра, матрица и список смежности, ориентированные и неориентированные графы.
11	Поиск в графах	5	BFS, DFS, кратчайший путь в невзвешенном графе, обнаружение связности и циклов.
12	Основы динамического программирования	5	перекрывающиеся подзадачи, мемоизация, табуляция, задачи на последовательности и оптимизацию.

Содержание модуля: Модуль дает системное представление об алгоритмах и структурах данных, которые используются при разработке backend-приложений, обработке данных, оптимизации запросов и проектировании сервисов. Практика строится на реализации алгоритмов на Python, тестировании решений и письменном объяснении выбора подхода.

Практические виды работ:

- реализация поиска, сортировок и структур данных на Python;
- решение задач на стек, очередь, словари, деревья и графы;
- сравнение эффективности решений на тестовых наборах данных;
- оформление решения с описанием сложности и ограничений.

Самостоятельная работа: решение задач на образовательной платформе, оформление кода в репозитории, самостоятельная доработка решений после проверки, подготовка кратких объяснений выбранного алгоритма.

Форма текущего и промежуточного контроля: автоматизированная проверка задач, код-ревью, тестирование по теории и промежуточный зачет.

Проектные задания модуля:

- **Key-Value хранилище на дереве поиска** – реализация собственной структуры хранения данных с операциями добавления, поиска, обновления и удаления записей;
- подготовка тестовых наборов данных и краткого объяснения сложности основных операций.

2.3.8. Модуль 8. Основы HTML/CSS/JavaScript

Цель модуля: сформировать у слушателя базовое понимание клиентской части веб-приложений, необходимое для разработки Django-шаблонов, форм, страниц личного кабинета и взаимодействия с backend/API.

Объем модуля: 56 часов.

Планируемые результаты по модулю:

- создавать семантическую HTML-разметку страниц, форм и таблиц;
- оформлять интерфейсы с применением CSS, адаптивной сетки и компонентов;
- использовать базовые конструкции JavaScript и обработчики событий;
- понимать цикл HTTP-запроса, обмен данными с API и роль frontend в backend-проекте.

Тематический план модуля:

Таблица 12 – Тематический план модуля 8 «Основы HTML/CSS/JavaScript»

№	Тема	Часы	Краткое содержание
1	Основы веба и HTTP	6	клиент и сервер, URL, HTTP-методы, статические и динамические страницы, инструменты разработчика браузера.

№	Тема	Часы	Краткое содержание
2	HTML: структура и семантика	9	структура документа, семантические теги, ссылки, изображения, списки, таблицы, доступность базового уровня.
3	HTML-формы	6	поля ввода, кнопки, атрибуты форм, валидация на стороне клиента, связь форм с backend-обработкой.
4	CSS: селекторы и каскад	8	подключение стилей, селекторы, наследование, специфичность, боксовая модель, единицы измерения.
5	Адаптивная верстка и Bootstrap	8	flex/grid на базовом уровне, медиа-запросы, сетка и компоненты Bootstrap, единый стиль интерфейса.
6	JavaScript: основы языка и DOM	8	переменные, функции, условия, массивы, события, изменение DOM, базовая интерактивность.
7	Работа с API из браузера	6	fetch, JSON, асинхронные запросы, отображение данных, обработка ошибок ответа.
8	Интеграция с backend-шаблонами	5	подготовка шаблонов для Django, статические файлы, наследование шаблонов, практический мини-проект страницы.

Содержание модуля: Модуль обеспечивает необходимую для backend-специалиста базу по устройству веб-страницы, формам, стилям, клиентской интерактивности, API-запросам и шаблонам. Практическая работа направлена на создание аккуратных интерфейсов для Django-проектов.

Практические виды работ:

- создание набора HTML-шаблонов для учебного web-приложения;
- оформление формы, таблицы и страницы списка объектов с помощью CSS/Bootstrap;
- добавление простых JS-обработчиков и запроса к API;
- подготовка шаблонов для последующей интеграции с Django.

Самостоятельная работа: выполнение мини-проекта интерфейса, оформление результата в репозитории, доработка по чек-листу доступности, адаптивности и корректности отображения.

Форма текущего и промежуточного контроля: проверка мини-проекта интерфейса, практический зачет по модулю.

Проектные задания модуля:

- **Веб-приложение прогноза погоды** – разработка страницы с выбором города, запросом к внешнему API и понятным отображением погодных данных;
- подготовка frontend-шаблонов для дальнейшей интеграции с Django-приложениями.

2.3.9. Модуль 9. Linux

Цель модуля: сформировать уверенные пользовательские и базовые административные навыки работы в среде Linux, необходимые backend-разработчику.

Объем модуля: 44 часов.

Планируемые результаты по модулю:

- ориентироваться в файловой системе Linux и использовать командную строку;
- управлять файлами, правами доступа и базовыми системными сервисами;
- писать простые Bash-скрипты;

- готовить рабочую серверную среду для запуска приложений, включая Docker, сервер приложений и базовые сценарии CI/CD.

Тематический план модуля:

Таблица 13 – Тематический план модуля 9 «Linux»

№	Тема	Часы	Краткое содержание
1	ИТ-системы и инфраструктура	9	роль операционной системы, сервера, сети и инфраструктуры в жизненном цикле программного продукта.
2	ОС Linux	8	командная строка, файловая система, команды навигации и работы с файлами.
3	Управление доступом и безопасностью	8	пользователи, группы, права доступа, безопасная работа с системой и секретами проекта.
4	Администрирование и переменные окружения	8	пакетный менеджер, службы, процессы, сетевые настройки, переменные окружения и журналы.
5	Docker и воспроизводимый запуск	6	контейнер, образ, Dockerfile, docker compose, пробный запуск Python/Django-проекта.
6	Bash, сервер приложений и CI/CD	5	Bash-скрипты, Gunicorn/Uvicorn, Nginx как reverse проху, базовая логика CI/CD-пайплайна.

Содержание модуля: В модуле рассматриваются ИТ-системы и место Linux в инфраструктуре разработки, базовые команды shell, управление правами доступа, основные административные операции, переменные окружения,

Docker, серверы приложений и автоматизация повторяющихся действий с помощью Bash. Практические задания ориентированы на навигацию в файловой системе, настройку окружения проекта и выполнение типовых операций, связанных с деплоем и сопровождением приложений.

Практические виды работ:

- работа в терминале и файловой системе Linux;
- настройка прав доступа и переменных окружения;
- создание Bash-скриптов для автоматизации действий;
- подготовка окружения для запуска Python-приложения;
- сборка простого Docker-образа, описание переменных окружения и подготовка схемы запуска через сервер приложений.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: проверка практических заданий в консоли и зачетный кейс по Linux.

Проектные задания модуля: слушатель подготавливает Linux-окружение для запуска выбранного Python-проекта, оформляет команды установки зависимостей, переменные окружения, Dockerfile или инструкцию контейнерного запуска, базовые сервисные настройки и инструкцию для повторяемого запуска.

2.3.10. Модуль 10. PostgreSQL

Цель модуля: сформировать прикладные навыки проектирования и использования базы данных PostgreSQL в backend-проектах на Python.

Объем модуля: 68 часов.

Планируемые результаты по модулю:

- проектировать схему БД, использовать ключи, ограничения и нормализацию;
- писать сложные SQL-запросы, включая JOIN, подзапросы, CTE и оконные функции;
- понимать транзакции, процедуры, триггеры и основы оптимизации;

- подключать PostgreSQL к Python-приложениям через ORM и драйверы.

Тематический план модуля:

Таблица 14 – Тематический план модуля 10 «PostgreSQL»

№	Тема	Часы	Краткое содержание
1	Введение	13	обзор предметной области, критерии качества, место модуля в общей траектории.
2	Ключи и DDL&DML	6	первичные и внешние ключи, ограничения, создание и изменение таблиц.
3	Отношения и нормализация	13	моделирование сущностей, связи, нормальные формы и устранение избыточности.
4	Транзакции	13	ACID, транзакционная обработка, фиксация и откат изменений.
5	Join и подзапросы	8	соединения таблиц, вложенные запросы и построение сложной выборки.
6	CTE & Window Functions	4	общие табличные выражения, оконные функции, аналитические запросы.
7	Функции & Процедуры	4	серверная логика, пользовательские функции, процедуры и их вызов.
8	Триггеры и оптимизация	8	триггеры, индексы, планы выполнения и базовые подходы к оптимизации.

№	Тема	Часы	Краткое содержание
9	Создание и подключение	8	инициализация сервера PostgreSQL, подключение клиентов, базовая настройка.
10	SQLAlchemy	5	ORM-подход, описание моделей, сессии и взаимодействие Python-приложения с БД.

Содержание модуля: Модуль строится вокруг практического освоения PostgreSQL: DDL и DML, ключи, ограничения, нормализация, транзакции, соединения таблиц, оконные функции, функции и процедуры, триггеры, индексы и основы оптимизации запросов. Отдельный блок посвящен подключению базы к Python-приложениям и использованию ORM-подхода. Практические задания выполняются на учебной БД, приближенной к реальным backend-сценариям.

Практические виды работ:

- проектирование и нормализация учебной базы данных;
- написание и оптимизация SQL-запросов различной сложности;
- создание функций, процедур и триггеров;
- подключение PostgreSQL к Python-проекту.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: практический зачет по проектированию схемы БД и набору SQL-задач.

Проектные задания модуля:

- перенос выбранного проекта с файлового или учебного хранения данных на PostgreSQL;

- подготовка схемы базы данных для проектов **Telegram-магазин с каталогом и корзиной, Django-приложение интернет-магазина, Интернет-магазин с регистрацией и корзиной и Продакшн-версия интернет-магазина на Django**;
- подключение Python-приложения к базе данных с использованием SQLAlchemy или средств Django ORM.

2.3.11. Модуль 11. Создание Telegram-бота

Цель модуля: сформировать навыки разработки прикладных Telegram-ботов на Python с использованием современного фреймворка и внешних сервисов.

Объем модуля: 56 часов.

Планируемые результаты по модулю:

- проектировать и реализовывать архитектуру Telegram-бота;
- использовать команды, клавиатуры, FSM и обработчики событий;
- интегрировать бота с базой данных и WebApp-интерфейсом;
- организовывать структуру проекта и сопровождение кода.

Тематический план модуля:

Таблица 15 – Тематический план модуля 11 «Создание Telegram-бота»

№	Тема	Часы	Краткое содержание
1	Введение	12	обзор предметной области, критерии качества, место модуля в общей траектории.
2	aiogram	12	архитектура aiogram, маршрутизация событий, обработчики сообщений и команд.
3	Кнопки	9	reply и inline-клавиатуры, пользовательская навигация и сценарии взаимодействия.

№	Тема	Часы	Краткое содержание
4	Машина состояний	9	FSM, пошаговые диалоги и валидация данных пользователя.
5	Структура проекта	9	разделение проекта на слои, конфигурации, хранилища и точки входа.
6	WebApp	5	встраиваемое web-приложение в Telegram, интеграция frontend и backend.

Содержание модуля: В модуле слушатели изучают жизненный цикл Telegram-бота, работу с фреймворком aiogram, команды и клавиатуры, конечные автоматы состояний, проектную структуру, а также подключение WebApp и внешних сервисов. Практика выстроена вокруг разработки рабочего чат-бота, выполняющего полезную бизнес-функцию: регистрацию пользователей, прием данных, обработку запросов и взаимодействие с БД.

Практические виды работ:

- реализация Telegram-бота с командами и клавиатурами;
- построение FSM-сценариев общения с пользователем;
- интеграция с базой данных и внешним API;
- создание мини-приложения WebApp для Telegram.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: практический модульный проект и зачет.

Проектные задания модуля:

- **Telegram-бот для управления задачами** – создание бота для добавления задач, просмотра списка дел и отметки выполненных пунктов;

- **Telegram-бот с прогнозом погоды** – интеграция Telegram-интерфейса с внешним погодным API;
- **Telegram-бот для бронирования столиков** – реализация сценария выбора даты, времени и количества гостей для ресторанов и кафе;
- **Telegram-магазин с каталогом и корзиной** – разработка каталога товаров, карточек, корзины и логики оформления заказа.

2.3.12. Модуль 12. Django

Цель модуля: сформировать у слушателя комплексные навыки серверной веб-разработки на фреймворке Django и подготовки приложения к развертыванию.

Объем модуля: 104 часов.

Планируемые результаты по модулю:

- создавать Django-проекты и приложения, проектировать модели и миграции;
- использовать Django admin, шаблоны, формы, авторизацию и представления на основе классов;
- разрабатывать API и организовывать middleware, сессии и пользовательский интерфейс;
- применять базовые меры безопасности web-приложений, хранить секреты вне кода и писать тесты;
- выполнять базовый деплой Django-приложения с использованием PostgreSQL, Docker, сервера приложений и CI/CD-подхода.

Тематический план модуля:

Таблица 16 – Тематический план модуля 12 «Django»

№	Тема	Часы	Краткое содержание
1	Как работает интернет	9	модель клиент-сервер, запрос и ответ, HTTP, URL и доменная адресация.

№	Тема	Часы	Краткое содержание
2	Сети	8	базовые сетевые протоколы, IP-адресация, порты, маршрутизация и сетевое взаимодействие приложений.
3	Веб-сервер и сервер приложений	8	веб-сервер, reverse проху, статика, WSGI/ASGI, Gunicorn/Uvicorn, связь с backend-приложением.
4	Django введение	8	структура проекта Django, настройки, маршрутизация и жизненный цикл запроса.
5	Модели	8	ORM, описание сущностей, связи, миграции, базовые операции с объектами.
6	Django-admin	8	административный интерфейс, настройка моделей, форм и представлений администратора.
7	DTL	8	шаблонизатор Django, наследование шаблонов, контекст, вывод и оформление данных.
8	Авторизация, регистрация и безопасность	8	пользовательская модель, формы входа и регистрации, разграничение доступа, CSRF, хранение секретов, проверка ввода.

№	Тема	Часы	Краткое содержание
9	Представления на основе классов	6	общие и специализированные представления на основе классов, повторное использование логики.
10	Миксины	6	переиспользование поведения, авторизационные и прикладные миксины.
11	API	6	проектирование API, сериализация данных, маршруты, основы Django REST Framework/FastAPI и взаимодействие клиента с сервером.
12	Сессия	6	механизм сессий, cookies, пользовательское состояние и безопасное хранение данных.
13	Middleware	6	роль middleware, порядок обработки запроса и ответа, прикладные сценарии расширения.
14	Деплой, Docker и CI/CD	9	конфигурация, переменные окружения, PostgreSQL, Docker, Gunicorn/Uvicorn, Nginx и базовый CI/CD-сценарий.

Содержание модуля: Модуль объединяет знания из предыдущих блоков и переводит их в полноценную backend-разработку на Django. Рассматриваются HTTP и сетевые основы, веб-сервер, проектная архитектура Django, моде-

ли, миграции, административный интерфейс, шаблонизатор DTL, авторизация и регистрация, представления на основе классов, миксины, API, безопасность, сессии, middleware, тестирование, Docker и деплой. Практическая часть завершается разработкой учебного веб-приложения с базой данных, пользовательским интерфейсом и серверной логикой.

Практические виды работ:

- создание Django-проекта с несколькими приложениями;
- реализация CRUD-функциональности и пользовательской аутентификации;
- разработка REST/API-слоя и интеграции с БД;
- написание базовых тестов и проверка критических пользовательских сценариев;
- подготовка приложения к развертыванию через переменные окружения, Docker, сервер приложений и reverse proxy.

Самостоятельная работа: изучение видео- и текстовых материалов, выполнение заданий на образовательной платформе, оформление результатов в репозитории, анализ ошибок после проверки преподавателем, доработка и повторная сдача заданий при необходимости.

Форма текущего и промежуточного контроля: рубежный проект по Django и промежуточный зачет по критериям качества backend-приложения.

Проектные задания модуля:

- **Django-приложение интернет-магазина** – разработка каталога товаров, карточек, изображений и структуры базы данных;
- **Интернет-магазин с регистрацией и корзиной** – реализация авторизации пользователей, корзины, добавления и удаления товаров;
- **Продакшн-версия интернет-магазина на Django** – подготовка Django-проекта с PostgreSQL, административной панелью, серверной настройкой, Docker-запуском, базовым CI/CD, деплоем на Linux и архитектурой, приближенной к требованиям бизнеса.

2.4. Проектная практика и консультации

Проектная практика и консультации занимают 24 часа и направлены на поэтапную подготовку итогового аттестационного проекта. В этот период слушатель:

- выбирает тему проекта из перечня проектных работ Программы или согласует индивидуальную тему с преподавателем;
- формирует техническое задание, архитектуру решения, структуру репозитория и перечень пользовательских сценариев;
- проводит интеграцию Python-части, базы данных, Telegram-бота и/или web-приложения;
- готовит пояснительную записку, README, инструкции по запуску, демонстрационные сценарии, тестовые сценарии и материалы для защиты;
- проходит консультации по архитектуре, БД, качеству кода, тестированию, безопасности, деплою, Docker/CI/CD и подготовке к итоговой аттестации.

В рамках проектной практики слушатель может довести до уровня итогового аттестационного проекта одну из комплексных тем: Telegram-бот для бронирования столиков, Telegram-магазин с каталогом и корзиной, Django-приложение интернет-магазина, интернет-магазин с регистрацией и корзиной или продакшн-версию интернет-магазина на Django. По решению преподавателя допускается развитие более раннего учебного проекта до комплексного уровня за счет подключения базы данных, web-интерфейса, Telegram-интерфейса, документации, тестовых сценариев, базовых мер безопасности, Docker-запуска и развертывания.

3. Организационно-педагогические условия реализации программы

3.1. Общие условия реализации

Реализация Программы осуществляется в очной форме с применением электронного обучения и дистанционных образовательных технологий в электронной информационно-образовательной среде организации. Образовательный процесс включает самостоятельное изучение материалов, выполнение практических работ, участие в консультациях, прохождение текущего контроля, промежуточной аттестации, проектной практики и итоговой аттестации.

Организация обеспечивает возможность освоения Программы независимо от места нахождения слушателя при условии соблюдения требований к техническому оснащению, идентификации слушателя и фиксации результатов обучения. Конкретные сроки начала и завершения обучения, расписание консультаций и контрольных мероприятий устанавливаются приказом, расписанием и/или индивидуальным учебным графиком.

3.2. Материально-технические условия

Для реализации Программы организация обеспечивает:

- доступ к электронной образовательной платформе/личному кабинету слушателя;
- размещение учебных материалов, заданий, тестов, инструкций и методических рекомендаций;
- инструменты приема и проверки практических работ;
- средства проведения консультаций, видеоконференций и обмена сообщениями;
- средства фиксации результатов текущего контроля, промежуточной и итоговой аттестации;
- техническую и методическую поддержку слушателей.

Минимальные требования к техническому обеспечению со стороны слушателя:

- персональный компьютер или ноутбук с устойчивым доступом в Интернет;

- операционная система Windows, Linux или macOS;
- установленный Python актуальной стабильной версии, Git, PostgreSQL или доступ к учебной базе данных;
- по мере прохождения модулей: Docker Desktop или Docker Engine, средства работы с переменными окружения, сервер приложений Gunicorn/Uvicorn, Nginx или учебный аналог reverse proxy, сервис CI/CD либо учебная имитация пайплайна;
- среда разработки: Visual Studio Code, PyCharm Community Edition или аналог;
- web-браузер, средства видеосвязи и возможность передачи файлов/ссылок на репозиторий;
- при необходимости – терминал, WSL/виртуальная машина, Docker или облачная среда разработки.

3.3. Электронная информационно-образовательная среда

Электронная информационно-образовательная среда используется для организации, сопровождения, контроля и документирования образовательного процесса. Она должна обеспечивать:

- доступ слушателей к учебному плану, календарному учебному графику, рабочим программам, учебным материалам, заданиям и оценочным средствам;
- размещение онлайн-курсов, видеоуроков, текстовых материалов, примеров кода, шаблонов проектов и инструкций;
- фиксацию факта доступа к материалам, сроков выполнения заданий, результатов тестирования, практических работ и аттестаций;
- хранение цифрового портфолио слушателя: выполненных заданий, ссылок на репозитории, замечаний преподавателя, результатов доработки и итогового проекта;
- взаимодействие слушателей с преподавателями, методистами и технической поддержкой;
- идентификацию и аутентификацию слушателя при проведении контрольных мероприятий в порядке, установленном локальными нормативными актами организации;

- фиксацию попыток прохождения контроля, сроков сдачи, результатов проверки, замечаний преподавателя и факта повторной сдачи;
- применение видеонаблюдения, прокторинга, проверки устным собеседованием или иных способов подтверждения самостоятельности при итоговой аттестации, если это предусмотрено локальным актом и доведено до слушателя до начала контрольного мероприятия;
- защиту персональных данных и разграничение прав доступа.

Допускается использование совокупности цифровых сервисов: LMS/образовательной платформы, системы видеоконференцсвязи, корпоративной почты/мессенджера, Git-репозитория, системы тестирования и сервиса хранения учебных материалов. Конкретные наименования сервисов, способы идентификации слушателя, правила аутентификации, порядок фиксации результатов и условия применения прокторинга утверждаются организацией и доводятся до слушателей до начала обучения. Использование ЕСИА, единой биометрической системы или иных государственных сервисов идентификации допускается только при наличии правовых и технических оснований и в порядке, установленном законодательством Российской Федерации и локальными актами организации.

3.4. Кадровые условия

Реализация Программы обеспечивается педагогическими работниками и (или) специалистами-практиками, квалификация которых соответствует профилю преподаваемых модулей. Преподаватель, эксперт или проверяющий практические работы должен иметь:

- высшее образование и/или среднее профессиональное образование по профилю информационных технологий, программной инженерии, прикладной математики, информатики либо документально подтвержденную профессиональную переподготовку по соответствующему профилю;
- практический опыт разработки программного обеспечения, web/backend-разработки, работы с Python, Git, SQL/PostgreSQL, Django или смежными технологиями;
- навыки применения электронного обучения и дистанционных образовательных технологий;

- готовность обеспечивать обратную связь, проверку практических работ и сопровождение итогового проекта.

При привлечении специалистов-практиков организация подтверждает их компетентность документами об образовании, сведениями об опыте работы, портфолио проектов, трудовыми договорами/договорами гражданско-правового характера или иными документами, предусмотренными локальными актами.

3.5. Учебно-методическое обеспечение

Учебно-методическое обеспечение Программы включает:

- утвержденную программу профессиональной переподготовки, учебный план, календарный учебный график и рабочие программы модулей;
- электронные учебные материалы: видеоуроки, текстовые конспекты, презентации, демонстрационные репозитории, инструкции, чек-листы и шаблоны проектов;
- оценочные материалы для текущего контроля, промежуточной и итоговой аттестации;
- методические рекомендации по выполнению практических заданий и итогового проекта;
- перечень рекомендуемой литературы, официальной документации и справочных ресурсов;
- инструкции по работе в электронной образовательной среде, сдаче заданий и коммуникации с преподавателем.

3.6. Применение электронного обучения и дистанционных образовательных технологий

Программа реализуется с применением электронного обучения и дистанционных образовательных технологий. Электронное обучение используется для предоставления учебных материалов, заданий, записей, методических рекомендаций, средств коммуникации, фиксации результатов контроля и обратной связи. Дистанционные образовательные технологии используются для организации взаимодействия слушателя и педагогических работников на рассто-

янии, проведения консультаций, проверки работ, промежуточной и итоговой аттестации.

При реализации Программы организация обеспечивает идентификацию слушателя в электронной информационно-образовательной среде, доступ к материалам в течение периода обучения, возможность сдачи работ, получение обратной связи и фиксацию результатов освоения Программы. Технические требования к устройству, программному обеспечению, каналам связи и учетным данным доводятся до слушателя до начала обучения или при предоставлении доступа к образовательной среде.

3.7. Информационная безопасность, персональные данные и этика

При реализации Программы обеспечивается соблюдение требований законодательства о персональных данных, правил работы с учетными записями, учебными материалами, проектными файлами и результатами аттестации. Доступ к электронной информационно-образовательной среде предоставляется индивидуально и не подлежит передаче третьим лицам.

Слушатель обязан соблюдать академическую добросовестность, не представлять чужие работы как собственные, корректно указывать источники заимствований, не нарушать права третьих лиц на программный код, данные, изображения, тексты и иные материалы. При выявлении признаков несамостоятельного выполнения работы организация вправе назначить дополнительную проверку, собеседование, повторную сдачу или иные процедуры, предусмотренные локальными нормативными актами.

3.8. Локальные процедуры обучения

Организация реализации Программы обеспечивается локальными нормативными актами ООО «Иннопрог», регуливающими прием и зачисление слушателей, режим занятий, применение электронного обучения и дистанционных образовательных технологий, текущий контроль, промежуточную и итоговую аттестацию, пересдачи, апелляции, отчисление, восстановление, выдачу документов о квалификации, внесение сведений в ФИС ФРДО и обработку персональных данных.

Локальные процедуры применяются в части, не противоречащей законодательству Российской Федерации, условиям договора об образовании и настоящей Программе. Слушатель информируется о значимых правилах реализации Программы до начала обучения или при предоставлении доступа к образовательной среде.

3.9. Обновление программы

Программа подлежит актуализации при изменении законодательства Российской Федерации в сфере образования, профессиональных стандартов, требований к применению электронного обучения и дистанционных образовательных технологий, а также при существенном изменении технологического стека Python/backend-разработки. Пересмотр содержания модулей, оценочных средств и методических материалов осуществляется не реже одного раза в три года либо чаще по решению организации.

4. Оценка качества освоения программы

4.1. Общие положения

Оценка качества освоения Программы проводится в формах текущего контроля успеваемости, промежуточной аттестации и итоговой аттестации. Оценочные материалы направлены на проверку достижения планируемых результатов обучения и сформированности профессиональных компетенций, заявленных в Программе.

Результаты контроля и аттестации фиксируются в электронной информационно-образовательной среде и/или в документах учета, предусмотренных локальными нормативными актами организации. При необходимости слушателю предоставляется возможность доработки практических заданий и повторной сдачи в порядке, установленном организацией.

4.2. Текущий контроль успеваемости

Текущий контроль проводится в ходе изучения каждого модуля и включает:

- автоматизированное тестирование по темам модуля;
- проверку практических заданий, задач и мини-проектов;
- код-ревью решений и анализ структуры репозитория;
- проверку соблюдения сроков, требований к оформлению и самостоятельности выполнения;
- обратную связь преподавателя, исправление ошибок и повторную сдачу доработанных решений.

Основные критерии текущего контроля:

- соответствие решения поставленной задаче;
- корректность, полнота и работоспособность результата;
- качество кода, читаемость, декомпозиция и соблюдение изученных подходов;
- наличие понятной структуры проекта, файла зависимостей, инструкции по запуску и осмысленной истории Git;
- наличие тестовых сценариев там, где это предусмотрено заданием;

- отсутствие секретов, токенов и паролей в открытом коде;
- умение объяснить принятые технические решения;
- самостоятельность выполнения и корректное оформление результата.

4.3. Промежуточная аттестация

Промежуточная аттестация проводится по завершении модулей и учитывает специфику содержания: тест, практическое задание, кейсовая работа, мини-проект, проверка репозитория или их комбинация. По промежуточной аттестации применяется оценка «зачтено / не зачтено». Для практических работ используется единый чек-лист: работоспособность, соответствие заданию, качество кода, структура проекта, Git, инструкция по запуску, обработка ошибок, тестовые сценарии при наличии такого требования и самостоятельность выполнения.

Таблица 17 – Формы промежуточной аттестации

Форма промежуточной аттестации	Что проверяется	Условие зачета
Тестирование	знание основных понятий, синтаксиса, терминов, последовательности действий	не менее 70% правильных ответов либо иной порог, установленный заданием
Практическое задание	способность применить знания для решения прикладной задачи	работоспособное решение без критических ошибок
Мини-проект	комплексное применение инструментов модуля	выполнены обязательные требования технического задания
Код-ревью / проверка репозитория	качество кода, структура, история изменений, оформление	репозиторий доступен, код запускается, замечания устранены

Отметка «зачтено» выставляется, если слушатель выполнил обязательные задания, продемонстрировал достижение результатов модуля, представил работоспособное решение и устранил критические замечания после проверки. Отметка «не зачтено» выставляется при невыполнении обязательного минимума заданий, отсутствии работоспособного результата, неподтвержденной самостоятельности выполнения или несоблюдении требований к оформлению.

4.4. Пересдача, апелляция и подтверждение самостоятельности

Слушатель, получивший результат «не зачтено» по текущему контролю или промежуточной аттестации, вправе устранить замечания и представить работу повторно в сроки, установленные преподавателем и локальным актом организации. При повторной сдаче проверяется исправление указанных ошибок и сохранение работоспособности всего решения.

Апелляция по результатам промежуточной или итоговой аттестации подается в порядке и сроки, установленные локальным нормативным актом организации. Рассмотрение апелляции проводится с учетом материалов проверки, версии работы в репозитории, комментариев проверяющего, демонстрации результата и пояснений слушателя.

Самостоятельность выполнения подтверждается историей Git, промежуточными версиями, пояснениями слушателя, устной защитой ключевых решений и проверкой совпадений при необходимости. Использование открытых источников, библиотек и фрагментов кода допускается при корректном указании источников и самостоятельной интеграции в проект.

4.5. Допуск к итоговой аттестации

К итоговой аттестации допускается слушатель, который:

- освоил все модули Программы;
- получил отметку «зачтено» по предусмотренным промежуточным аттестациям;
- выполнил практические задания и проектную практику в объеме, установленном учебным планом;
- не имеет академической задолженности;

- представил комплект материалов итогового проекта в установленные сроки.

4.6. Итоговая аттестация

Итоговая аттестация является обязательной и проводится в форме **защиты итогового аттестационного проекта**. Итоговый проект представляет собой самостоятельную разработку прикладного программного продукта, выполненную на языке Python с использованием инструментов, изученных в Программе. Защита проводится дистанционно с применением средств видеосвязи и/или образовательной платформы.

4.6.1. Примерная тематика проектных работ

Перечень проектных работ используется для текущего контроля, промежуточной аттестации, проектной практики и итоговой аттестации. Для итоговой аттестации тема должна быть доведена до комплексного уровня: иметь прикладную цель, структуру кода, хранение данных, пользовательский сценарий, документацию и демонстрацию работоспособности. Примерный перечень тем приведен ниже.

Таблица 18 – Примерная тематика проектных работ

№	Проект	Содержание проектной работы
1	Система учета фруктов на складе	Консольная программа для учета товаров на складе: добавление позиций, контроль остатков, обновление количества и формирование понятного отчета по доступной продукции.
2	База зарплат сотрудников	Система для хранения данных о сотрудниках, управления зарплатами и быстрого поиска информации, направленная на автоматизацию базовых HR- и бухгалтерских операций.
3	Менеджер задач в консоли	Приложение для постановки, отслеживания и закрытия задач, позволяющее управлять личными или рабочими делами в одном месте.

№	Проект	Содержание проектной работы
4	Telegram-бот для управления задачами	Telegram-бот для добавления задач, просмотра списка дел и отметки выполненных пунктов прямо в мессенджере.
5	Сервис бронирования поездок	Система для оформления поездок, учета клиентов и транспорта, моделирующая работу небольшого сервиса заказа машин и автоматизацию бронирования.
6	Сервис управления пассажирскими перевозками	Приложение для учета транспорта, вместимости и пассажиров, моделирования посадки, контроля загрузки и отслеживания доступных мест.
7	Складской менеджер товаров	Система управления ассортиментом магазина с учетом остатков, цен и доступности товаров, поддерживающая быстрый поиск позиций и контроль склада.
8	Key-Value хранилище на дереве поиска	Собственная система хранения данных с добавлением, поиском, обновлением и удалением записей на основе дерева поиска.
9	Веб-приложение прогноза погоды	Современный сайт для отображения актуальной погоды по выбранному городу с использованием внешнего API, удобного интерфейса и понятной визуализации данных.
10	Telegram-бот с прогнозом погоды	Бот, который по запросу пользователя показывает прогноз погоды и демонстрирует работу с внешним API и Telegram-интерфейсом.
11	Telegram-бот для бронирования столиков	Бот для ресторанов и кафе, через который клиенты выбирают дату, время и количество гостей, а бизнес принимает и обрабатывает заявки.

№	Проект	Содержание проектной работы
12	Telegram-магазин с каталогом и корзиной	Telegram-бот с каталогом товаров, карточками, корзиной и логикой оформления заказа для продаж в мессенджере.
13	Django-приложение интернет-магазина	Интернет-магазин на Django с каталогом товаров, карточками, изображениями и структурированной базой данных для управления ассортиментом.
14	Интернет-магазин с регистрацией и корзиной	Е-commerce проект с авторизацией пользователей, корзиной, добавлением и удалением товаров, приближенный к коммерческой разработке.
15	Продакшн-версия интернет-магазина на Django	Веб-проект на Django с PostgreSQL, административной панелью, серверной настройкой, Docker-запуском, базовым CI/CD, деплоем на Linux и архитектурой, приближенной к требованиям бизнеса.

Для итоговой аттестации предпочтительными являются комплексные темы 11–15. Темы 1–10 могут быть использованы как основа итогового проекта при условии согласованного усложнения: добавления базы данных, Telegram-интерфейса или web-интерфейса, тестовых сценариев, документации и развертывания.

4.6.2. Обязательные требования к итоговому проекту

Итоговый проект должен:

- быть выполнен преимущественно на языке Python;
- иметь понятную архитектуру, декомпозицию и прикладную бизнес-логику;
- использовать постоянное хранение данных: PostgreSQL, SQLite для учебного прототипа или иную согласованную базу данных;
- включать пользовательский сценарий взаимодействия через web-интерфейс, Telegram-интерфейс, API или их комбинацию;
- иметь репозиторий с исходным кодом и историей разработки;

- содержать файл README с инструкцией по запуску, требованиями к окружению, переменными окружения и описанием основных сценариев;
- содержать пояснительную записку с описанием цели, архитектуры, моделей данных, API или интерфейсов, мер безопасности, ограничений и направлений развития;
- демонстрировать самостоятельность выполнения и соответствие требованиям академической добросовестности;
- содержать тестовые сценарии или описание ручной проверки критического функционала;
- для комплексных проектов предусматривать воспроизводимый запуск через инструкции, Docker или иной согласованный способ.

4.6.3. Порядок проведения итоговой аттестации

Итоговая аттестация включает:

1. предоставление слушателем ссылки на репозиторий и комплекта материалов итогового проекта;
2. предварительную проверку комплектности и работоспособности проекта;
3. демонстрацию работы проекта в дистанционном формате;
4. устную защиту архитектуры, выбранных технологий, логики работы, базы данных и ограничений решения;
5. ответы на вопросы комиссии/эксперта;
6. принятие решения об успешном или неуспешном прохождении итоговой аттестации.

4.6.4. Критерии оценивания итогового проекта

Таблица 19 – Критерии оценивания итогового аттестационного проекта

Критерий	Макс. балл
Соответствие проекта заявленной теме, цели и техническому заданию	10

Работоспособность и полнота реализованного функционала	18
Качество архитектуры решения, декомпозиция и обоснованность выбранных технологий	14
Качество Python-кода, читаемость, структурированность, соблюдение стиля	10
Корректность работы с базой данных, моделями данных и миграциями/схемой	10
Тестирование, обработка ошибок, логирование и безопасность конфигурации	8
Использование Git, структура репозитория и история разработки	5
Наличие документации по запуску, использованию, переменным окружения и тестированию проекта	10
Качество демонстрации и ответов на вопросы при защите	8
Самостоятельность выполнения и отсутствие критических заимствований	7
Итого	100

Итоговая аттестация считается успешно пройденной, если слушатель набрал не менее 70 баллов и не получил нулевую оценку по критически значимым критериям: «работоспособность», «архитектура» и «самостоятельность».

Критические основания для неудовлетворительного результата итоговой аттестации:

- проект не запускается и не может быть проверен;
- отсутствует исходный код или обязательные материалы проекта;
- заявленная функциональность фактически не реализована;
- выявлены признаки несамостоятельного выполнения без указания источников;
- слушатель не может объяснить архитектуру, код и принятые решения.

4.7. Внутренняя и внешняя оценка качества реализации программы

Внутренняя оценка качества реализации Программы включает:

- анализ результатов текущего контроля, промежуточной и итоговой аттестации;
- экспертную проверку актуальности рабочих программ модулей и оценочных материалов;
- анализ типовых ошибок слушателей и корректировку учебных материалов;
- сбор обратной связи от слушателей, преподавателей и проверяющих;
- анализ качества итоговых проектов и соответствия результатов заявленным компетенциям;
- проверку корректности ведения электронных журналов, портфолио, протоколов аттестации и материалов итоговой защиты;
- документирование изменений и актуализацию Программы.

Внешняя независимая оценка качества может проводиться через привлечение профильных работодателей, специалистов-практиков, независимых экспертов, представителей профессионального сообщества или организаций, имеющих опыт подготовки и оценки ИТ-специалистов. Внешняя оценка может включать экспертизу программы, рецензирование оценочных материалов, участие эксперта в итоговой защите, анализ обезличенных итоговых проектов и рекомендации по актуализации содержания.

5. Методические материалы

5.1. Методические указания для слушателей

Слушателю рекомендуется:

- изучать материалы модулей последовательно, не пропуская темы;
- выполнять практические задания сразу после изучения теории;
- фиксировать изменения и версии проекта в Git-репозитории;
- документировать ключевые решения и замечания по коду;
- регулярно использовать среду разработки, терминал, систему контроля версий и базу данных как единый рабочий контур;
- обращаться к официальной документации технологий;
- начинать подготовку итогового проекта заранее и вести его поэтапно на протяжении программы.

5.2. Методические указания для преподавателей и проверяющих

Преподавателю рекомендуется:

- обеспечивать практико-ориентированный характер занятий;
- использовать единые критерии проверки кода и проектов;
- уделять внимание корректности решения и качеству инженерной реализации;
- обеспечивать своевременную обратную связь по заданиям;
- организовывать модульные проверки таким образом, чтобы они формировали задел для итогового проекта;
- актуализировать примеры и кейсы с учетом современных практик Python-разработки.

5.3. Рекомендуемая литература и ресурсы

1. Официальная документация Python. Python Software Foundation.
2. PEP 8 - Style Guide for Python Code.
3. Luciano Ramalho. Fluent Python. O'Reilly Media.
4. Al Sweigart. Automate the Boring Stuff with Python.

5. Mark Lutz. Learning Python.
6. Pro Git. Scott Chacon, Ben Straub.
7. Официальная документация PostgreSQL.
8. Alan Beaulieu. Learning SQL.
9. Aditya Bhargava. Grokking Algorithms.
10. MDN Web Docs: HTML, CSS, JavaScript.
11. Официальная документация Django.
12. Официальная документация aiogram.
13. William Shotts. The Linux Command Line.
14. Официальная документация Docker.
15. Официальная документация Django REST Framework.
16. Официальная документация FastAPI.
17. OWASP Web Security Testing Guide и OWASP Top 10 как справочные материалы по безопасности web-приложений.

Приложение 1. Матрица формирования компетенций

Таблица 20 – Матрица формирования компетенций

Модуль	ПК-1	ПК-2	ПК-3	ПК-4	ПК-5	ПК-6	ПК-7	ПК-8	ПК-9	ПК-10
Python начальный	+									
Python продвинутый	+	+								
ООП в Python	+	+								
Git			+							
Основы SQL				+						
Алгоритмы и структуры данных	+				+					
Основы HTML/CSS/JavaScript						+				
Linux							+			
PostgreSQL				+				+		
Создание Telegram-бота	+	+	+	+				+	+	
Django	+	+	+	+		+		+		+
Проектная практика и итоговая аттестация	+	+	+	+	+	+	+	+	+	+

Приложение 2. Оценочные материалы промежуточной аттестации

Промежуточная аттестация проводится в форме зачета по соответствующему модулю. Конкретный вариант задания утверждается организацией и размещается в электронной информационно-образовательной среде. Оценка выставляется по системе «зачтено / не зачтено». Минимальное условие зачета: выполнение обязательной части задания, отсутствие критических ошибок, подтверждаемая самостоятельность выполнения и соблюдение требований к оформлению результата.

Таблица 21 – Оценочные материалы промежуточной аттестации

№	Модуль	Форма контроля	Компетенции	Пример зачетного задания	Критерии и порог зачета
1	Введение в профессию и организация обучения	тест и проверка настройки рабочей среды	организационная готовность, ПК-3, ПК-7	пройти входной тест, войти в личный кабинет, настроить IDE, Git, терминал и создать учебный репозиторий	не менее 70% теста, доступы работают, репозиторий создан, инструкция по сдаче работ понята
2	Python начальный	набор задач с автоматической и ручной проверкой	ПК-1	реализовать консольную программу учета товаров или задач с вводом, условиями, циклами, функциями и отчетом	не менее 70% задач выполнено, обязательная часть запускается, базовые ошибки синтаксиса отсутствуют
3	Python продвинутый	практикум по файловой утилите и модульной структуре проекта	ПК-1, ПК-2	разработать файловую утилиту для чтения CSV/JSON, обработки ошибок, логирования, сохранения результата и базовых pytest-тестов	обязательная часть без критических ошибок, есть структура модулей, зависимости, инструкция запуска и тестовые сценарии
4	ООП в Python	практическая работа по проектированию классов	ПК-2	спроектировать объектную модель сервиса перевозок или склада с классами, методами, инкапсуляцией и валидацией состояния	не менее 70 баллов из 100, модель соответствует предметной области, код читаем и объясним

Продолжение таблицы 21

№	Модуль	Форма контроля	Компетенции	Пример зачетного задания	Критерии и порог зачета
5	Git	кейс по работе с репозитори- ем	ПК-3	создать репозиторий, офор- мить ветку задачи, выполнить несколько осмысленных ком- митов, слияние, README и описание изменений	все обязательные операции выполнены, история Git ло- гична, конфликт или слияние обработаны корректно
6	Основы SQL	SQL-практикум	ПК-4	создать таблицы учебной предметной области, запол- нить данные, написать вы- борки с фильтрацией, JOIN, группировкой и изменением записей	не менее 70% запросов верны, структура БД соответствует заданию, опасные операции подтверждены проверкой результата
7	Алгоритмы и структуры данных	задачи на алгоритмы с объяс- нением сложности	ПК-1, ПК-5	реализовать Key-Value хра- нилище на дереве поиска и решить набор задач на спис- ки, словари, стек, очередь или сортировку	не менее 70% задач выполне- но, решения проходят тесты, сложность и выбор структуры данных объяснены
8	Основы HTML/CSS/JavaScript	мини-проект интерфейса	ПК-6	сверстать страницу погодного приложения с формой, адап- тивной разметкой, базовой интерактивностью и запросом к внешнему API	обязательные элементы реа- лизованы, интерфейс открыва- ется в браузере, JS-логика работает, ошибки ответа API обработаны
9	Linux	практический кейс по ко- мандной строке и запуску приложения	ПК-7	подготовить окружение про- екта: команды установки, права доступа, переменные окружения, Bash-скрипт, Dockerfile или инструкцию контейнерного запуска	все критические команды выполнены, проект воспроиз- водимо запускается, секреты не размещены в открытом коде

Продолжение таблицы 21

№	Модуль	Форма контроля	Компетенции	Пример зачетного задания	Критерии и порог зачета
10	PostgreSQL	практикум по базе данных	ПК-4, ПК-8	спроектировать схему БД для магазина или бронирования, добавить ограничения, связи, индексы, запросы и подключение из Python/Django	не менее 70% задания выполнено, связи и ограничения корректны, приложение подключается к БД
11	Создание Telegram-бота	мини-проект бота	ПК-1, ПК-3, ПК-4, ПК-8, ПК-9	разработать Telegram-бота для задач, погоды, бронирования или магазина с командами, клавиатурами, хранением данных и обработкой ошибок	бот запускается, обрабатывает основные сценарии, данные сохраняются, README содержит инструкцию и переменные окружения
12	Django	мини-проект web-приложения	ПК-1, ПК-2, ПК-3, ПК-4, ПК-6, ПК-8, ПК-10	разработать Django-приложение интернет-магазина с моделями, миграциями, шаблонами, авторизацией, формами, тестами и базовым деплоем	обязательный функционал реализован, БД подключена, критические сценарии проверены, секреты вынесены из кода, запуск описан
13	Проектная практика	проверка этапов итогового проекта	ПК-1–ПК-10	согласовать тему, подготовить техническое задание, архитектуру, репозиторий, прототип, тестовые сценарии и демонстрационный сценарий	этапы сданы в срок, замечания устранены, проект готов к предварительной проверке перед защитой

Единый чек-лист проверки практических работ:

- решение соответствует заданию и запускается у проверяющего по инструкции;
- структура проекта понятна, файлы названы осмысленно, зависимости зафиксированы;
- код декомпозирован, читаем, не содержит дублирования сверх необходимого и соответствует изученным подходам;
- ошибки пользователя, пустые данные и нештатные сценарии обрабатываются;

- данные сохраняются в согласованном формате или базе данных, если это требуется заданием;
- в репозитории есть осмысленная история коммитов и актуальный README;
- токены, пароли, ключи API и иные секреты не размещены в открытом коде;
- тесты, тестовые данные или инструкция ручной проверки приложены, если они предусмотрены заданием;
- слушатель может объяснить архитектуру, ключевые функции и причины выбранных технических решений.

Критические ошибки, приводящие к результату «не зачтено»:

- работа не запускается и не может быть проверена по приложенной инструкции;
- отсутствует обязательная часть задания или ключевой пользовательский сценарий;
- отсутствует исходный код, репозиторий или требуемые материалы сдачи;
- в открытый репозиторий помещены реальные токены, пароли или секретные ключи;
- выполнены опасные операции с данными без проверки и обоснования, например массовое удаление без условий в учебной БД;
- выявлены признаки несамостоятельного выполнения, которые слушатель не может объяснить на собеседовании;
- работа нарушает требования локального акта об академической добросовестности.

Порядок пересдачи и апелляции: слушатель получает замечания по работе, устраняет их и повторно размещает исправленный результат в электронной информационно-образовательной среде. Сроки, количество попыток, формат повторной проверки и порядок апелляции устанавливаются локальными нормативными актами организации. При повторной сдаче слушатель обязан сохранить историю доработки и быть готовым объяснить внесенные изменения.

Приложение 3. Оценочные материалы итоговой аттестации

Форма итоговой аттестации: защита итогового аттестационного проекта.

Примерные темы итоговых аттестационных проектов и проектных работ:

1. Система учета фруктов на складе: программа для учета товаров, контроля остатков, обновления количества и формирования отчета по продукции.
2. База зарплат сотрудников: система хранения данных о сотрудниках, управления зарплатами и быстрого поиска информации.
3. Менеджер задач в консоли: приложение для постановки, отслеживания и закрытия задач.
4. Telegram-бот для управления задачами: бот для добавления задач, просмотра списка дел и отметки выполненных пунктов.
5. Сервис бронирования поездок: система оформления поездок, учета клиентов и транспорта.
6. Сервис управления пассажирскими перевозками: приложение для учета транспорта, вместимости, пассажиров и доступных мест.
7. Складской менеджер товаров: система управления ассортиментом магазина, остатками, ценами и доступностью товаров.
8. Key-Value хранилище на дереве поиска: собственная структура хранения данных с добавлением, поиском, обновлением и удалением записей.
9. Веб-приложение прогноза погоды: сайт с выбором города, внешним API и отображением погодных данных.
10. Telegram-бот с прогнозом погоды: бот для получения прогноза погоды через Telegram-интерфейс.
11. Telegram-бот для бронирования столиков: бот для выбора даты, времени и количества гостей в ресторане или кафе.
12. Telegram-магазин с каталогом и корзиной: бот с карточками товаров, корзиной и логикой оформления заказа.
13. Django-приложение интернет-магазина: web-приложение с каталогом товаров, карточками, изображениями и базой данных.

14. Интернет-магазин с регистрацией и корзиной: e-commerce проект с авторизацией пользователей, корзиной, добавлением и удалением товаров.
15. Продакшн-версия интернет-магазина на Django: Django-проект с PostgreSQL, административной панелью, серверной настройкой, Docker-запуском, базовым CI/CD, деплоем на Linux и архитектурой, приближенной к требованиям бизнеса.

Комплексные темы 11–15 рекомендуются как основные темы итоговой аттестации. Темы 1–10 могут быть использованы для итоговой аттестации после согласованного усложнения и включения постоянного хранения данных, пользовательского интерфейса, документации и демонстрации работоспособности.

Минимальные требования к комплекту материалов итоговой аттестации:

- ссылка на репозиторий с исходным кодом;
- файл README с инструкцией по запуску;
- описание структуры проекта;
- файл с зависимостями и/или конфигурацией окружения;
- схема БД либо описание используемых моделей данных;
- краткая пояснительная записка по архитектуре решения;
- демонстрационные сценарии использования.

Рекомендуемая структура пояснительной записки к итоговому проекту:

1. тема, цель и назначение проекта;
2. постановка задачи и основные пользовательские сценарии;
3. архитектура и состав компонентов;
4. используемый стек технологий;
5. описание моделей данных и хранилища;
6. описание интерфейсов взаимодействия (web, Telegram, API);
7. инструкция по запуску и тестированию;
8. ограничения и направления дальнейшего развития.

Основания для недопуска к защите проекта:

- отсутствие обязательных материалов проекта;
- отсутствие работоспособной версии решения;

- отсутствие подтверждаемой самостоятельной работы;
- академическая задолженность по учебному плану.

Форма оценочного листа итоговой защиты:

Таблица 22 – Оценочный лист итоговой аттестации

№	Проверяемый показатель	Балл	Комментарий эксперта
1	Соответствие теме, цели и техническому заданию	0–10	
2	Работоспособность и полнота функционала	0–18	
3	Архитектура, декомпозиция и выбор технологий	0–14	
4	Качество Python-кода и структура проекта	0–10	
5	Работа с базой данных, моделями и миграциями/схемой	0–10	
6	Тестирование, логирование, обработка ошибок и безопасность конфигурации	0–8	
7	Git, репозиторий и история разработки	0–5	
8	Документация по запуску, использованию и тестированию	0–10	
9	Демонстрация и ответы на вопросы	0–8	
10	Самостоятельность выполнения	0–7	

Продолжение таблицы 22

№	Проверяемый показатель	Балл	Комментарий эксперта
Итого		0–100	

Решение по итоговой аттестации: итоговая аттестация считается пройденной при результате не менее 70 баллов, отсутствии критических оснований для неудовлетворительной оценки и подтвержденной самостоятельности выполнения проекта.

Приложение 4. Перечень локальных нормативных актов, на которые должна опираться реализация программы

При введении Программы в действие организация обеспечивает наличие, актуальность и применение локальных нормативных актов, регламентирующих образовательную деятельность по дополнительным профессиональным программам, в том числе:

1. порядок разработки, согласования, утверждения, актуализации и хранения дополнительных профессиональных программ;
2. положение об организации образовательного процесса по дополнительным профессиональным программам;
3. положение о применении электронного обучения и дистанционных образовательных технологий;
4. положение об электронной информационно-образовательной среде;
5. положение о текущем контроле успеваемости, промежуточной и итоговой аттестации слушателей;
6. порядок допуска к итоговой аттестации и оформления результатов итоговой аттестации;
7. порядок подачи и рассмотрения апелляций по результатам промежуточной и итоговой аттестации;
8. положение о порядке приема, зачисления, перевода, отчисления, восстановления и завершения обучения слушателей;
9. формы договора об образовании, заявления на обучение, согласия на обработку персональных данных и иных документов приема;
10. положение об оказании платных образовательных услуг;
11. положение о языке образования;
12. порядок учета, хранения, заполнения и выдачи документов о квалификации и справок об обучении;
13. порядок передачи сведений о выданных документах о квалификации в ФИС ФРДО;
14. положение о защите персональных данных слушателей и работников;
15. положение об академической добросовестности и предотвращении неправомерных заимствований;

16. инструкции для слушателей по работе в электронной образовательной среде, сдаче заданий и прохождению аттестации;
17. порядок идентификации, аутентификации и прокторинга при проведении контрольных мероприятий с применением электронного обучения и дистанционных образовательных технологий;
18. должностные инструкции/функциональные обязанности преподавателей, экспертов, методистов, кураторов и технической поддержки;
19. формы приказов об утверждении программы, зачислении, отчислении, допуске к итоговой аттестации, завершении обучения и выдаче документов о квалификации.

До подачи документов для получения или переоформления лицензии рекомендуется проверить соответствие указанных локальных актов фактической модели реализации Программы, сведениям о педагогических работниках, материально-техническому обеспечению, электронной информационно-образовательной среде и правилам выдачи документов о квалификации.

Приложение 5. Лист актуализации программы

Лист актуализации используется для фиксации изменений Программы «Python-разработчик», связанных с обновлением законодательства, профессиональных стандартов, квалификационных требований, учебных материалов, оценочных средств, цифровой образовательной среды и проектной практики.

Таблица 23 – Лист актуализации программы

№	Дата	Основание изменения	Содержание изменения
1	01.05.2026	Первичное утверждение	Программа введена в действие приказом организации.
2			
3			

Приложение 6. Требования к репозиторию, README и демонстрации

Для подтверждения освоения Программы и выполнения итогового проекта слушатель представляет проектные материалы в форме, позволяющей проверить самостоятельность, воспроизводимость и качество результата.

Минимальный состав проектных материалов:

- репозиторий или архив проекта с исходным кодом и структурой каталогов;
- файл README.md или иной документ с описанием цели, функций, структуры, установки и запуска проекта;
- описание используемых технологий, зависимостей, версии языка/фреймворка/инструментов;
- инструкции по подготовке окружения, настройке переменных, запуску приложения, тестов или демонстрационного сценария;
- демонстрационные данные, скриншоты, ссылка на развернутую версию или видеодемонстрация при наличии такой возможности;
- краткое описание ограничений, известных проблем и возможных направлений развития проекта.

Материалы должны быть достаточны для повторной проверки проекта преподавателем, экспертом или членом аттестационной комиссии без дополнительных устных пояснений, за исключением вопросов по защите.

Приложение 7. Дорожная карта подготовки выпускного проекта

Дорожная карта используется как методический ориентир для поэтапной подготовки итогового аттестационного проекта по квалификации «Python-разработчик».

1. Выбор темы и согласование прикладной задачи с преподавателем или куратором.
2. Формулирование цели проекта, целевой аудитории, пользовательских сценариев и критериев приемки.
3. Подготовка структуры проекта, репозитория, базовой документации и плана реализации.
4. Реализация минимально работоспособной версии и фиксация результата в репозитории.
5. Расширение функциональности, обработка ошибок, подготовка данных, интерфейса или демонстрационного сценария в соответствии со спецификой направления.
6. Проверка качества, тестирование, рефакторинг, оформление инструкций по запуску и использованию.
7. Предзащита или консультационная проверка, устранение замечаний, подготовка презентации и демонстрации.
8. Итоговая защита проекта с ответами на вопросы комиссии и фиксацией результата итоговой аттестации.

Приложение 8. Контрольный лист соответствия требованиям программы

Контрольный лист применяется для внутренней проверки полноты Программы и готовности ее реализации.

Таблица 24 – Контрольный лист соответствия требованиям программы

№	Проверяемый элемент	Статус	Комментарий
1	Указаны вид программы, квалификация, трудоемкость, форма обучения, язык и срок освоения.	выполнено	
2	Представлены учебный план, календарный учебный график и рабочие программы модулей.	выполнено	
3	Описаны планируемые результаты, профессиональные компетенции и связь с квалификационными ориентирами.	выполнено	
4	Определены организационно-педагогические условия, ЭИОС, ЭО и ДОТ, кадровые и методические условия.	выполнено	
5	Описаны текущий контроль, промежуточная аттестация, итоговая аттестация, передача и апелляция.	выполнено	

Продолжение таблицы 24

№	Проверяемый элемент	Статус	Комментарий
6	Определены документы по итогам обучения и перечень локальных актов, обеспечивающих реализацию.	выполнено	