

ПРИКАЗ № Python/2025

об утверждении дополнительной профессиональной программы

01.01.2025 г., г. Санкт-Петербург

В соответствии с Федеральным законом № 273-ФЗ от 29.12.2012 «Об образовании в Российской Федерации» (ст. 12, 75, 101), Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам» приказываю:

1. Утвердить дополнительную профессиональную программу – программу профессиональной переподготовки «Профессия: Python-разработчик» общей трудоемкостью 253,2 академ. ч.
2. Контроль исполнения приказа оставляю за собой.



/ Зинин Е.Ю.

Утверждено
Приказом № Python/2025 от 01.01.2025 г.
Генеральный директор
Зинин Е.Ю.
«01» января 2025 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
– ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ
«ПРОФЕССИЯ: PYTHON-РАЗРАБОТЧИК»**

Срок реализации: 8 месяцев
Количество часов: 253,2 акад. ч.
Форма обучения: очно-заочная форма
Формат обучения: с применением
исключительно дистанционных технологий
Возраст обучающихся: для лиц старше 18
лет, имеющих или получающих среднее
профессиональное и (или) высшее
образование
Автор: Микрюкова Е.Р.

г. Санкт-Петербург
2025

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая дополнительная профессиональная программа – программа профессиональной переподготовки «Профессия: Python-разработчик» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Постановлением Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи"»;
- Профессиональным стандартом 06.001 «Программист», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. № 424н;
- ФГОС высшего образования – магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника, утв. приказом Минобрнауки России от 19.09.2017 г. № 918.

В данной программе учтены основные идеи формирования универсальных учебных действий учащихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

Направленность программы:

Программа имеет техническую направленность.

Адресат:

Программа рассчитана на обучающихся старше 18 лет, имеющих или получающих среднее профессиональное и (или) высшее образование.

Актуальность реализации:

Современный рынок информационных технологий испытывает высокий спрос на специалистов, владеющих языком программирования Python, который используется в веб-разработке, анализе данных, машинном обучении и автоматизации бизнес-процессов. Программа направлена на подготовку профессионалов, способных решать практические задачи, востребованные в IT-компаниях и стартапах. Особое внимание уделяется формированию прикладных навыков разработки, работе с современными инструментами и технологиями. Завершив обучение, выпускники смогут создавать полнофункциональные приложения и интегрировать их в реальную бизнес-среду. Реализация программы способствует повышению конкурентоспособности специалистов на рынке труда.

Отличительные особенности программы:

Программа построена на практико-ориентированном подходе и включает работу над реальными проектами, приближенными к задачам IT-компаний. Содержание курса охватывает все ключевые этапы разработки: от базового синтаксиса Python до создания

полнофункциональных приложений и работы с современными фреймворками. Обучение проходит в онлайн-формате с применением интерактивных материалов, домашних заданий и индивидуальной обратной связи от преподавателей. Особое внимание уделяется формированию навыков командной разработки, использованию Git и работе с системой контроля версий. Завершение программы позволяет слушателям сформировать портфолио проектов, востребованных на рынке труда.

Объем и срок освоения программы: 253,2 академ. ч. в течение 8 месяцев.

Доступ к материалам Программы обучающимся предоставляется на 3 года. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

Выдаваемый документ о квалификации:

Обучающиеся успешно прошедшие Программу и сдавшие итоговую аттестацию вправе получить диплом о профессиональной переподготовке установленного ООО «Тривиум» образца.

Цели и задачи программы:

Цель дополнительной профессиональной программы «Профессия: Python-разработчик» – подготовка специалистов, обладающих знаниями и практическими навыками разработки программных решений на языке Python, способных создавать, тестировать и внедрять современные приложения и сервисы.

Программа направлена на решение следующих основных задач:

Обучающие:

- Дать слушателям знания по синтаксису языка Python и основам алгоритмизации.
- Научить проектировать архитектуру приложений и использовать современные инструменты и фреймворки.
- Освоить основы работы с базами данных, API и внешними библиотеками.
- Сформировать умения писать чистый, оптимизированный и поддерживаемый код.
- Обучить методикам тестирования, отладки и деплоя приложений.

Развивающие:

- Развить аналитическое и алгоритмическое мышление.
- Сформировать навыки решения нестандартных инженерных задач.
- Развить умение работать с технической документацией и открытыми источниками.
- Содействовать развитию навыков командной разработки и взаимодействия через Git.
- Формировать способность быстро адаптироваться к новым инструментам и технологиям.

Воспитательные:

- Воспитать ответственное отношение к качеству создаваемого программного продукта.
- Сформировать культуру командной работы и уважения к коллегам.
- Развить профессиональную этику и дисциплину при работе над проектами.
- Воспитать стремление к саморазвитию и постоянному совершенствованию профессиональных навыков.
- Сформировать готовность принимать решения и нести ответственность за результат.

Планируемые результаты:

Планируемыми результатами обучения по программе является приобретение следующих знаний, умений, навыков, участвующих в качественном изменении компетенций:

Знания:

- Основы языка программирования Python, его синтаксис, структуры данных и ключевые возможности.
- Принципы объектно-ориентированного программирования и архитектурные подходы к построению приложений.
- Основы работы с базами данных, SQL и инструментами ORM.
- Концепции клиент-серверного взаимодействия, API и веб-разработки.
- Методики тестирования, отладки и оптимизации программного кода.

Умения:

- Создавать, тестировать и развёртывать приложения на Python.
- Разрабатывать архитектуру программных решений и использовать сторонние библиотеки и фреймворки.
- Работать с системами контроля версий (Git) и организовывать совместную разработку.
- Подключать базы данных, выполнять запросы и обрабатывать большие объёмы данных.
- Использовать инструменты автоматизации и интеграции приложений.

Навыки:

- Написание чистого, читаемого и поддерживаемого кода.
- Применение Python в реальных проектах и решение прикладных задач.
- Работа в команде разработчиков с использованием Agile-практик и Git-flow.
- Подготовка и оформление технической документации к проектам.
- Разработка полнофункциональных приложений, пригодных к внедрению в рабочую среду.

Перечень профессиональных компетенций, на получение которых направлено обучение:

На основе профстандарта «Программист»:

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации

А	Разработка и отладка программного кода	3	Формализация и алгоритмизация поставленных задач для разработки программного кода	A/01.3	3
			Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных	A/02.3	3
			Оформление программного кода в соответствии с установленными требованиями	A/03.3	3
			Работа с системой управления версиями программного кода	A/04.3	3
			Проверка и отладка программного кода	A/05.3	3
В	Проверка работоспособности и рефакторинг кода программного обеспечения	4	Разработка процедур проверки работоспособности и измерения характеристик компьютерного программного обеспечения	B/01.4	4
			Проверка работоспособности компьютерного программного обеспечения	B/03.4	4
			Рефакторинг, оптимизация и инспекция программного кода	B/04.4	4
			Исправление дефектов программного кода, зафиксированных в базе данных дефектов	B/05.4	4
			Осуществление сборки однородных программных модулей в программный проект	B/06.4	4

С	Интеграция программных модулей и компонентов и проверка работоспособности и выпусков программного продукта	5	Разработка процедур интеграции программных модулей	С/01.5	5
---	--	---	--	--------	---

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- способность формализовывать и алгоритмизировать поставленные задачи для последующей разработки программного кода на языке Python;
- умение разрабатывать и писать программный код с использованием современных инструментов и фреймворков, включая работу с базами данных;
- готовность оформлять программный код в соответствии с установленными стандартами и требованиями проекта;
- владение инструментами управления версиями и навыками командной разработки с использованием систем контроля версий;
- способность выполнять проверку, отладку, оптимизацию и рефакторинг кода для повышения качества программных решений;
- умение разрабатывать и применять процедуры тестирования, обеспечивающие работоспособность и надежность программного обеспечения;
- способность интегрировать программные модули и компоненты, проводить сборку приложений и проверку работоспособности финальных версий продукта.

Организационно-педагогические условия реализации программы дополнительного профессионального образования:

Язык реализации образовательной программы: обучение проводится на русском языке.

Форма обучения: очно-заочная форма.

Особенности реализации программы: программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

Условия набора: на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

Формы проведения занятий:

- занятия в видео-формате;
- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа;
- индивидуальные вопросы.

Материально-техническое оснащение:

Материальное обеспечение программы

Занятия проводятся на образовательной онлайн-платформе «Productstar». Каждый обучающийся и педагог оснащены доступом к образовательной онлайн-платформе: <https://platform.productstar.ru/login>.

У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

Методическое обеспечение программы

Программа обеспечена:

- учебно-методическими материалами;
- практическими заданиями и тренажерами;
- теоретическими и практическими видео-занятиями.

Кадровое обеспечение:

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

1) отвечающее одному из требований:

а) имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования «Образование и педагогические науки»;

б) имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Тривиум», и получение при необходимости дополнительного профессионального образования педагогической направленности;

в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности программы дополнительного профессионального образования;

2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;

3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

2. УЧЕБНЫЙ ПЛАН

№ п/п	Наименование блока	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1.	Основы Python	39,5	13,1	26,4	Текущий контроль, промежуточная аттестация
2.	Основы SQL	14,4	3,8	10,6	Текущий контроль
3.	SQL и базы данных	27,6	6,5	21,1	Текущий контроль, промежуточная аттестация
4.	Системы контроля версий. Работа с Flask	30,2	5,8	24,4	Текущий контроль, промежуточная аттестация
5.	ООП в Python	47	11,4	35,6	Текущий контроль, промежуточная аттестация
6.	Бэкенд-разработка на Python и Django	78,4	13,7	64,7	Текущий контроль, промежуточная аттестация
7.	Современные методологии разработки	3,3	2,4	0,9	Не предусмотрено
8.	Дипломная работа	12,8	1,6	11,2	Итоговая аттестация
	Итого:	253,2	58,3	194,9	

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование блока	1 месяц	2 месяц	3 месяц	4 месяц	5 месяц	6 месяц	7 месяц	8 месяц
1.	Основы Python	X	X						
2.	Основы SQL			X					
3.	SQL и базы данных				X				
4.	Системы контроля версий. Работа с Flask					X			
5.	ООП в Python						X		
6.	Бэкенд-разработка на Python и Django							X	X
7.	Современные методологии разработки								X
8.	Дипломная работа								X

4. РАБОЧАЯ ПРОГРАММА

Блок 1. Основы Python

Теория 13,1 академ. ч. Практика 26,4 академ. ч.

В рамках блока ставятся следующие задачи:

- Освоить основы синтаксиса Python и базовые конструкции языка.
- Научиться работать с переменными, типами данных, списками, словарями и функциями.
- Освоить применение условных операторов и циклов для решения практических задач.
- Получить навыки работы с библиотеками, модулями и структурами данных.
- Научиться отлаживать код и обрабатывать ошибки.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают основы языка Python, его синтаксис и базовые конструкции.
- Умеют применять функции, условные операторы, циклы и структуры данных.
- Владеют инструментами работы с библиотеками и модулями.
- Способны находить и исправлять ошибки в коде.
- Могут создавать небольшие программы и решать практические задачи.

Блок состоит из следующих тем:

Тема 1: Введение в Python

Тема 2: Практическая работа № 1: основы Python

Тема 3: Переменные и типы данных. Условные операторы

Тема 4: Строки, условия и циклы

Тема 5: Списки и словари в Python

Тема 6: Практическая работа № 2: основы Python

Тема 7: Функции

Тема 8: Практическая работа № 3: основы Python

Тема 9: Проект: разработка чат-бота на Python

Тема 10: Пакеты и модули. Pip

Тема 11: Библиотеки

Тема 12: Структуры данных в Python

Тема 13: Практическая работа № 4: основы Python

Тема 14: Ошибки и исключения в Python

Тема 15: Практическая работа № 5: основы Python

Тема 16: Аналитический проект на Python

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических заданий, направленных на закрепление ключевых тем блока: написание базового кода, работа с переменными и функциями, создание циклов, применение библиотек и модулей, отладка и исправление ошибок. Содержание и критерии практических заданий указываются в личном кабинете на образовательной онлайн-платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа выполнена частично, содержит ошибки или недочёты; допускается повторная сдача после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация 1: проект «Разработка чат-бота на Python». В рамках проекта слушатели применяют переменные, условные конструкции и циклы; создают и вызывают функции; работают со словарями, списками и файлами; отлаживают код и обрабатывают исключения. Решение загружается в личный кабинет, проверяется педагогами. Оценка выставляется в том же формате: «зачёт», «на доработку», «незачёт».

Промежуточная аттестация 2: аналитический проект на Python. В рамках проекта слушатели проводят исследовательский анализ данных; используют pandas для описания и группировки данных; применяют matplotlib и seaborn для визуализации результатов; составляют аналитический отчёт в Jupyter Notebook. Решение загружается в личный кабинет, проверяется педагогами. Оценка выставляется в том же формате: «зачёт», «на доработку», «незачёт».

Блок 2. Основы SQL

Теория 3,8 академ. ч. Практика 10,6 академ. ч.

В рамках блока ставятся следующие задачи:

- Освоить базовые концепции работы с реляционными базами данных и языком SQL.
- Научиться извлекать, фильтровать и сортировать данные с помощью операторов SQL.
- Получить навыки преобразования данных, работы с регистрами, арифметическими операциями и структурами CASE.
- Освоить группировку данных, агрегатные функции и сортировку значений.
- Применять знания SQL для решения практических задач и анализа данных.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают основные принципы работы реляционных баз данных, назначение SQL и сферы его применения.
- Умеют писать SQL-запросы для извлечения, фильтрации, сортировки и группировки данных.
- Владеют навыками использования агрегатных функций, операторов сравнения, логических условий и структуры CASE.
- Способны преобразовывать данные и выполнять вычисления в запросах.
- Готовы применять SQL для анализа данных и работы с реальными кейсами.

Блок состоит из следующих тем:

Тема 1: Введение в блок SQL

Тема 2: Основы извлечения и фильтрации данных

Тема 3: Расширенные возможности фильтрации и извлечения данных

Тема 4: Практическая работа № 1: основы SQL

Тема 5: Преобразование данных на практике

Тема 6: Преобразование даты и сортировка данных на практике

Тема 7: Группировка данных на практике

Тема 8: Практическая работа № 2: основы SQL

Форма контроля: текущий контроль

Текущий контроль осуществляется через выполнение практических заданий, направленных на закрепление ключевых тем блока: написание SQL-запросов для извлечения, фильтрации, сортировки и группировки данных, работа с арифметическими функциями, регистрами и структурами CASE.

Содержание и критерии практических заданий указываются в личном кабинете на образовательной онлайн-платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, запросы корректны и соответствуют заданию.
- На доработку — работа выполнена частично, содержит ошибки или недочёты; допускается повторная сдача после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Блок 3. SQL и базы данных

Теория 6,5 академ. ч. Практика 21,1 академ. ч.

В рамках блока ставятся следующие задачи:

- Освоить принципы работы с базами данных и их архитектурой.
- Научиться создавать, изменять и удалять таблицы и управлять их связями.
- Освоить методы извлечения, фильтрации, объединения и сортировки данных с помощью SQL-запросов.
- Получить навыки использования подзапросов, оконных функций и оптимизации запросов.
- Научиться работать с PostgreSQL, MySQL и инструментами администрирования баз данных.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают основы проектирования баз данных, типы запросов и их назначение.
- Умеют создавать таблицы, управлять структурами данных и использовать подзапросы.
- Владеют методами оптимизации запросов и работы с оконными функциями.
- Способны объединять и анализировать данные из нескольких таблиц.
- Могут применять PostgreSQL, MySQL и дополнительные инструменты для работы с БД на практике.

Блок состоит из следующих тем:

- Тема 1: Введение в базы данных
- Тема 2: Объединение таблиц
- Тема 3: Практическое применение подзапросов
- Тема 4: Практическая работа № 1: SQL и базы данных
- Тема 5: Обновление, добавление и удаление данных
- Тема 6: Создание, изменение и удаление таблиц
- Тема 7: Операции с текстом и оконные функции
- Тема 8: Практическая работа № 2: SQL и базы данных
- Тема 9: Ускорение и оптимизация запросов, табличные выражения
- Тема 10: Практическая работа №3: SQL и базы данных
- Тема 11: Проект от компании "ILine": онлайн-каталог сотрудников
- Тема 12: Визуализация данных

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических заданий по написанию SQL-запросов, работе с таблицами, объединению данных, применению подзапросов и оптимизации запросов. Содержание и критерии практических заданий указываются в личном кабинете на образовательной онлайн-платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа выполнена частично, содержит ошибки или недочёты; допускается повторная сдача после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация: проект от компании "Iline" — «Онлайн-каталог сотрудников». В рамках проекта слушатели проектируют схему базы данных, создают консольное приложение для управления информацией о сотрудниках, используют PostgreSQL, MySQL, Python, GitHub и библиотеку Mimesis. Решение загружается в личный кабинет, проверяется педагогами. Оценка выставляется в том же формате: «зачёт», «на доработку», «незачёт».

Блок 4. Системы контроля версий. Работа с Flask

Теория 5,8 академ. ч. Практика 24,4 академ. ч.

В рамках блока ставятся следующие задачи:

- Освоить принципы работы с системами контроля версий и основными командами Git.
- Научиться использовать GitHub для управления проектами, командной работы и решения конфликтов.
- Освоить работу с ветками, коммитами, репозиториями и продвинутыми командами Git.
- Познакомиться с архитектурой REST и принципами построения API.

- Научиться разрабатывать и тестировать API-сервисы на базе Flask.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают основные принципы работы систем контроля версий, команды Git и возможности GitHub.
- Умеют создавать, клонировать и администрировать репозитории, работать с ветками и коммитами.
- Владеют инструментами построения REST API и разработки приложений на Flask.
- Способны разрабатывать API-сервисы и интегрировать их с внешними приложениями.
- Могут создавать небольшие веб-сервисы и работать с запросами и ответами HTTP.

Блок состоит из следующих тем:

Тема 1: Системы контроля версий

Тема 2: Git & GitHub

Тема 3: Продвинутая работа с Git

Тема 4: Практическая работа № 1: системы контроля версий. Работа с Flask

Тема 5: Разработка API на Flask

Тема 6: Проект от компании «E-soft»: API-сервис для обработки данных на Flask

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических заданий, направленных на закрепление ключевых тем блока: работа с Git, создание и клонирование репозитория, управление коммитами, настройка API и разработка REST-сервисов. Содержание и критерии практических заданий указываются в личном кабинете на образовательной онлайн-платформе. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа выполнена частично, содержит ошибки или недочёты; допускается повторная сдача после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация: проект от компании «E-soft» — разработка API-сервиса на Flask для обработки данных. Слушатели создают два микросервиса: для хранения заметок и для работы с календарем, применяя принципы архитектуры веб-сервисов и REST API. Решение загружается в личный кабинет, проверяется педагогами. Оценка выставляется в том же формате: «зачёт», «на доработку», «незачёт».

Блок 5. ООП в Python

Теория 11,4 академ. ч. Практика 35,5 академ. ч.

В рамках блока ставятся следующие задачи:

- Освоить основные концепции объектно-ориентированного программирования на Python.
- Научиться создавать и использовать классы, объекты, методы и атрибуты.
- Изучить работу с дандер-методами и их применение в практике.
- Освоить принципы написания качественного кода: абстракция, композиция, инкапсуляция, SOLID.
- Получить опыт разработки проектов с использованием ООП и REST API.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают ключевые концепции ООП, принципы SOLID и лучшие практики написания кода.
- Умеют проектировать классы и объекты, использовать дандер-методы и реализовывать наследование.
- Владеют навыками работы с ORM SQLAlchemy и интеграции ООП с REST API.
- Способны разрабатывать приложения с использованием принципов ООП и тестировать их работоспособность.
- Могут применять функциональное программирование и advanced-функции в Python.

Блок состоит из следующих тем:

Тема 1: Объектно-ориентированное программирование на Python

Тема 2: Практическая работа № 1: ООП в Python

Тема 3: Дандер методы

Тема 4: Практическая работа № 2: ООП в Python

Тема 5: Свойства хорошего кода. Abstraction, Composition, Coupling, Cohesion

Тема 6: Принципы SOLID

Тема 7: Практическая работа № 3: ООП в Python

Тема 8: Функциональное программирование в Python

Тема 9: Advanced функции в Python

Тема 10: Проект от компании «ILine»: система управления персоналом

Тема 11: REST API Flask и сериализация

Тема 12: Практическая работа № 4: ООП в Python

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических заданий: написание классов, работа с дандер-методами, проектирование API и реализация принципов SOLID. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа выполнена частично, содержит ошибки или недочёты; допускается повторная сдача после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация: проект от компании «ILine» — «Система управления персоналом». В рамках проекта слушатели проектируют модели данных и создают схему БД с использованием SQLAlchemy; разрабатывают REST API для управления сотрудниками; реализуют работу с запросами и сериализацию данных; тестируют и отлаживают проект. Оценка выставляется по системе: «зачёт», «на доработку», «незачёт».

Блок 6. Бэкенд-разработка на Python и Django

Теория 13,7 академ. ч. Практика 64,7 академ. ч.

В рамках блока ставятся следующие задачи:

- Изучить архитектуру веб-приложений и основные принципы серверной разработки.
- Освоить работу с API, включая JSON, XML, AJAX и интеграцию с внешними сервисами.
- Научиться разрабатывать REST API с использованием Django и Django REST Framework.
- Освоить многопоточность, асинхронное программирование и масштабируемость приложений.
- Получить практический опыт развертывания проектов, настройки CI/CD и работы с Docker.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают архитектуру серверных приложений, работу API, принципы многопоточности и асинхронности.
- Умеют создавать REST API, работать с Django REST Framework, реализовывать аутентификацию и авторизацию.
- Владеют навыками интеграции бэкенда с внешними сервисами и сериализации данных.
- Способны разрабатывать масштабируемые приложения, настраивать CI/CD и автоматизировать тестирование.
- Могут развертывать проекты на удаленных серверах, использовать Docker, Nginx и Unicorn.

Блок состоит из следующих тем:

Тема 1: Архитектура и серверная часть

Тема 2: JSON, XML, AJAX и работа с API

Тема 3: Практическая работа № 1: бэкенд-разработка на Python и Django

Тема 4: Многопоточность

Тема 5: Асинхронное программирование

Тема 6: Практическая работа № 2: бэкенд-разработка на Python и Django

Тема 7: Проект от компании «Alpina Digital»: API конструктор ботов на основе GPT

Тема 8: Работа с ORM

Тема 9: Практическая работа № 3: бэкенд-разработка на Python и Django

Тема 10: Авторизация и аутентификация, OAuth 2.0

Тема 11: CI/CD

Тема 12: Практическая работа № 4: бэкенд-разработка на Python и Django

Тема 13: Масштабируемость

Тема 14: Тестирование в PyUnit

Тема 15: Практическая работа № 5: бэкенд-разработка на Python и Django

Тема 16: Создание Docker-образа для проекта на Django

Тема 17: Проект от компании «Alpina Digital»: развертывание проекта, CI/CD

Тема 18: Искусственный интеллект для разработчика

Форма контроля: текущий контроль, промежуточная аттестация

Текущий контроль осуществляется через выполнение практических заданий: разработку API, интеграцию с внешними сервисами, настройку Docker и CI/CD, тестирование приложений. Работы загружаются в личный кабинет и проверяются педагогами.

Оценивание осуществляется по системе:

- Зачёт — работа выполнена полностью, все основные критерии соблюдены, решение корректно и соответствует заданию.
- На доработку — работа выполнена частично, содержит ошибки или недочёты; допускается повторная сдача после корректировок.
- Незачёт — задание выполнено с критическими ошибками, не соответствует требованиям или не демонстрирует освоение материала. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

Промежуточная аттестация 1: проект от компании «Alpina Digital» — «API конструктор ботов на основе GPT». В рамках проекта слушатели создают REST API, используют Django REST Framework, пишут сериализаторы и тесты, настраивают аутентификацию и авторизацию, интегрируют API с внешними сервисами. Оценка выставляется по системе: «зачёт», «на доработку», «незачёт».

Промежуточная аттестация 2: проект от компании «Alpina Digital» — «Развертывание проекта, CI/CD». Слушатели настраивают сервер, готовят Docker-окружение, запускают проект с использованием Gunicorn и Nginx, организуют автоматизацию деплоя. Оценка выставляется по системе: «зачёт», «на доработку», «незачёт».

Блок 7. Современные методологии разработки

Теория 2,4 академ. ч. Практика 0,9 академ. ч.

В рамках блока ставятся следующие задачи:

- Изучить основные современные методологии управления разработкой программного обеспечения.
- Освоить различия между гибкими (Agile, Scrum) и классическими (Waterfall) подходами.
- Получить навыки применения инструментов планирования и коммуникации в команде.
- Научиться использовать практические элементы Scrum: создание user story, определение ролей, организация рабочих процессов.
- Ознакомиться с инструментами управления задачами и взаимодействия внутри команды.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают основные методологии управления разработкой, их преимущества и ограничения.
- Умеют применять гибкие подходы, разрабатывать user story и распределять задачи в команде.
- Владеют навыками использования инструментов для планирования, коммуникации и управления проектами (Slack, Microsoft Teams, таск-трекеры).
- Способны эффективно организовывать рабочее пространство и выстраивать процессы командного взаимодействия.

Блок состоит из следующих тем:

Тема 1: Методологии управления: Agile и Waterfall

Тема 2: Scrum

Тема 3: Инструменты управления

Форма контроля: не предусмотрено

Блок 8. Дипломная работа

Теория 1,6 академ. ч. Практика 11,2 академ. ч.

В рамках блока ставятся следующие задачи:

- Подготовить и оформить дипломную работу по выбранному проекту.
- Закрепить знания и навыки, полученные в ходе курса, на практике.
- Научиться структурировать проект, документировать его и презентовать результат.
- Освоить правила защиты диплома и представления проекта комиссии.

Планируемые результаты обучения:

По завершении блока слушатели:

- Знают основные критерии оценки дипломной работы, структуру проектной документации и подходы к её оформлению.
- Умеют разрабатывать, тестировать и документировать полноценные проекты на Python с использованием изученных технологий.
- Владеют навыками подготовки презентации проекта и его защиты.
- Способны продемонстрировать знания и умения, полученные в ходе всего курса, на примере итогового проекта.

Блок состоит из следующих тем:

Тема 1: Дипломная работа

Тема 2: Рост в тимлида

Финальное задание является формой итоговой аттестации и направлено на комплексную проверку освоения всех ключевых тем программы. Для выполнения слушателям предлагается выбрать один из вариантов дипломных проектов:

1. Проект от компании «ILine»: система управления персоналом.
2. Проект от компании «E-soft»: API-сервис для обработки данных на Flask.
3. Проект от компании «Alpina Digital»: развертывание проекта через CI/CD.

Дополнительно допускается защита собственного рабочего проекта, выполненного на Python.

Оценивается:

1. Качество кода: читаемость, эффективность, наличие комментариев, использование стандартов и подходящих алгоритмов.
2. Функциональность: реализованные возможности соответствуют требованиям.
3. Документация: описание архитектуры, функционала, способов использования и принципов работы проекта.
4. Понимание решений: умение объяснить логику кода и выбранные подходы.
5. Оригинальность: самостоятельное написание кода и понимание его работы.
6. Презентация: корректное оформление и наличие активной ссылки на проект.

Решение направляется на проверку через личный кабинет. Оценивание осуществляется по системе:

- Зачёт — все критерии выполнены, приложение работает стабильно, структура проекта оформлена корректно.
- Доработка — задание содержит несущественные ошибки или неточности, возможно повторное представление после исправлений.
- Незачёт — критические нарушения, задание не демонстрирует достаточный уровень освоения программы. После двух попыток доработки педагог вправе выставить итоговую оценку «незачёт».

В случае успешной сдачи итоговой аттестации обучающийся получает диплом о профессиональной переподготовке установленного образца.

5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Программа обеспечена образовательной онлайн-платформе:
<https://platform.productstar.ru/login>.

Педагогические технологии:

- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

Дидактический материал:

1. «Изучаем Python», Марк Лутц.
2. «Простой Python», Билл Любанович.
3. «Python. Книга рецептов», Дэвид Бизли, Брайан К. Джонс.
4. «Python. Разработка на основе тестирования», Гарри Персиваль.

Электронно-библиотечные ресурсы и системы, информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы, размещённые в личном кабинете обучающегося.

Оценочные материалы:

Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль, промежуточный контроль и итоговое оценивание.

Текущий контроль

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося на образовательной онлайн-платформе. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если текущий контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическом режиме.

Примеры тестовых вопросов

1. **Основы Python.** Какой тип данных в Python является изменяемым?
 - a) tuple
 - б) str
 - в) list
 - г) int
2. **SQL.** Какая конструкция используется для объединения строк в одну с разделителем?
 - a) GROUP BY
 - б) CONCAT
 - в) JOIN
 - г) CASE
3. **Git.** Какая команда создаёт новую ветку и сразу переключает на неё?
 - a) git branch new-feature
 - б) git checkout new-feature
 - в) git checkout -b new-feature
 - г) git merge new-feature
4. **ООП.** Какой принцип ООП отвечает за «скрытие деталей реализации»?
 - a) Наследование
 - б) Инкапсуляция
 - в) Полиморфизм
 - г) Абстракция
5. **Django.** Какой инструмент используется для построения REST API в Django?
 - a) Flask
 - б) Django ORM

- в) Django REST Framework
- г) SQLAlchemy

Примеры домашних заданий

1. **Python** (блок 1):
Напишите программу, которая принимает на вход строку и выводит:

- количество слов,
- количество уникальных слов,
- самое длинное слово.

2. **SQL** (блок 2–3):
Используя базу данных «магазин», напишите SQL-запросы:

- вывести все заказы за последний месяц;
- найти ТОП-3 товара по количеству продаж;
- посчитать средний чек по каждому пользователю.

3. **Flask** (блок 4):
Создайте простое API на Flask с двумя эндпоинтами:

- /add_note — добавление заметки (POST, принимает текст заметки),
 - /get_notes — получение всех заметок (GET).
- Данные храните в словаре или списке.

Промежуточный контроль

Промежуточный контроль проводится после изучения следующих блоков: 1, 3-6. Тематика и условия выполнения работ в рамках промежуточного контроля расписаны в личном кабинете обучающегося на образовательной онлайн-платформе. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическом режиме на образовательной онлайн-платформе.

Примеры проектных работ

Проект 1. «Онлайн-каталог сотрудников» (блок 3 — SQL и базы данных)

Цель: Научиться проектировать базу данных, управлять таблицами и интегрировать SQL с Python.

Описание задания:

- Спроектировать схему базы данных для хранения информации о сотрудниках (ФИО, должность, дата приёма, зарплата, отдел).
- Создать таблицы в PostgreSQL или MySQL.
- Реализовать консольное приложение на Python для:
 1. Добавления нового сотрудника.
 2. Обновления информации о сотруднике.
 3. Удаления сотрудника.

4. Поиска сотрудников по отделу и должности.

- Использовать библиотеку SQLAlchemy или стандартный модуль sqlite3.
- В репозитории на GitHub оформить README с инструкцией по запуску.

Проект 2. «API-сервис для заметок» (блок 4 — Flask и Git)

Цель: Освоить основы веб-разработки на Flask, REST API и работу с системой контроля версий.

Описание задания:

- Создать приложение на Flask, которое реализует API для работы с заметками.
- Функционал API:
 1. Добавление заметки (POST).
 2. Получение списка всех заметок (GET).
 3. Получение заметки по ID (GET).
 4. Удаление заметки (DELETE).
- Реализовать хранение данных в памяти (словарь/список) или в SQLite.
- Организовать работу с GitHub: вести репозиторий, сделать не менее 3 осмысленных коммитов, оформить README.md.

Итоговое оценивание

В конце программы обучающиеся сдают итоговую аттестацию. Для успешного прохождения итогового контроля слушатель должен выполнить критерии, указанные в разделе 4.

Результаты текущего контроля, промежуточной аттестации и итогового оценивания отображаются в личном кабинете слушателя на образовательной онлайн-платформе: <https://platform.productstar.ru/login>.

По результатам сдачи текущего контроля, промежуточного контроля и итогового оценивания педагог даёт обратную связь слушателям, отмечает их сильные стороны и обращает внимание на зоны для развития. При необходимости педагог может повторить пройденные темы со слушателями, если установлен факт плохого закрепления и усвоения темы у слушателей.