

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

Институт управления
Факультет менеджмента

«УТВЕРЖДАЮ»
Директор Института управления
М.В Хайруллина

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
повышения квалификации
«Python-разработчик»**

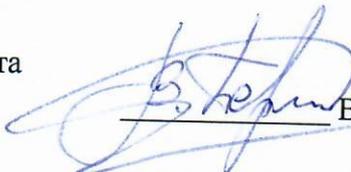
Москва, 2026

Разработчики:

Старший преподаватель кафедры статистики и анализа данных
факультета менеджмента
Института управления

 Н.И. Пышков

Кандидат экономических наук,
доцент кафедры управления проектами
и предпринимательства факультета менеджмента
Института управления

 В.В. Перцов

Руководитель программы:

Старший преподаватель кафедры статистики и анализа данных
факультета менеджмента
Института управления

 Н.И. Пышков

**Руководитель
структурного подразделения:**

Заместитель директора
Института управления

 Н.В. Демидова

Программа повышения квалификации рассмотрена на заседании Ученого совета
Института управления и рекомендована к реализации, протокол № 3 от «24» 02
2026 г.,

Выписка из протокола № 3
внеочередного заочного заседания Ученого совета Института управления
федерального государственного бюджетного образовательного
учреждения высшего образования
«Российская академия народного хозяйства и государственной службы
при Президенте Российской Федерации»

от 24 февраля 2026 года

По пункту 2 повестки дня

Представлены материалы: служебная записка от 19.02.2026 № 1522-26-24-СЗ об утверждении дополнительной профессиональной программы повышения квалификации «Python-разработчик», реализуемой в рамках национального проекта «Кадры».

Итоги голосования: «ЗА» 22, «ПРОТИВ» 0, «ВОЗДЕРЖАЛОСЬ» 0.

Решение заседания: по итогам заочного голосования («за» единогласно) принято решение утвердить дополнительную профессиональную программу повышения квалификации «Python-разработчик».

Председатель Ученого совета
директор Института управления



М.В. Хайруллина

И.о. ученого секретаря



И.В. Гунина

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ.....	4
1.1. Цель и задачи реализации программы.....	4
1.2. Нормативные правовые акты.....	4
1.3. Планируемые результаты обучения	5
1.4. Категория слушателей	8
1.5. Формы и технологии обучения.....	8
1.6. Период обучения, срок освоения и режим занятий	8
1.7. Документ о квалификации.....	8
2. СОДЕРЖАНИЕ ПРОГРАММЫ	9
2.2 Учебный план	10
2.3. Содержание программы повышения квалификации	11
3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ	14
3.1. Материально-техническое и программное обеспечение реализации программы.....	14
3.2 Учебно-методическое и информационное обеспечение программы....	15
4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ.....	20
5. ИНДИКАТОРЫ СФОРМИРОВАННЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКА ПРОГРАММЫ.....	30

Приложение 1. Рецензии (внешняя и внутренняя)

Приложение 2. Кадровая справка

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель и задачи реализации программы

Программа направлена на переобучение и профессиональное образование с возможностью дальнейшего трудоустройства в российские организации, освоение новой компетенции и применение полученных умений и навыков на практике.

Цель реализации программы: Основной целью программы является ознакомление слушателей с основными возможностями языка Python и его применение при разработке программ.

Задачи реализации программы:

- рассмотрение областей современной разработки программного обеспечения (веб-приложения, анализ данных, системное администрирование, и тд).
- формирование понимания конструкций языка Python, типов и структуры данных, функций, объектно-ориентированного и функционального подходов к программированию;
- освоение практических навыков разработки и реализации несложных алгоритмов на языке программирования Python.

1.2. Нормативные правовые акты

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

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Минобрнауки России от 24 марта 2025 г. № 266 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам».
3. Постановление Правительства РФ от 11 октября 2023 г. № 1678 "Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ.
4. «Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов» (утв. Минобрнауки России 22.01.2015 № ДЛ-1/05вн).
5. Приказ Министерства образования науки Российской Федерации от 23.08.2017 № 816 (зарегистрирован Минюстом России 18.09.2017, регистрационный № 48226). «Об утверждении порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных

- образовательных технологий при реализации образовательных программ.
6. Приказ Минобрнауки России от 19.09.2017 N 922 "Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика"
 7. Профессиональный стандарт 06.001 Программист, утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. N 424н;
 8. Постановление Правительства Российской Федерации от 13.03.2021 года №362 "О государственной поддержке в 2021 году юридических лиц и индивидуальных предпринимателей при трудоустройстве безработных граждан".
 9. Приказ РАНХиГС от 22 сентября 2017 года №01-6230 «Об утверждении Положения о применении в Академии электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
 10. Приказ РАНХиГС от 19 апреля 2019 г. № 02-461 «Об утверждении локальных нормативных актов РАНХиГС по дополнительному профессиональному образованию» (с изменениями и дополнениями);
 11. Приказ РАНХиГС от 02 декабря 2025 года №02-02669/001 «Об утверждении порядка разработки и утверждения в РАНХиГС дополнительных профессиональных программ - программ повышения квалификации, программ профессиональной переподготовки».
 12. Приказ РАНХиГС «Об утверждении Правил приема на обучение по дополнительным профессиональным программам в Академию» №02-00010/001 от 13 января 2026 года.
 13. Приказ РАНХиГС «Об утверждении Положения об итоговой аттестации слушателей дополнительных профессиональных программ в Академии» №02-00009/001 от 13 января 2026 года.
 14. Приказ РАНХиГС от 22 сентября 2017 года №01-6230 «Об утверждении Положения о применении в Академии электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».

1.3. Планируемые результаты обучения

Выпускник по программе повышения квалификации «Python-разработчик», в соответствии с задачами профессиональной деятельности и целями программы должен обладать следующими основными компетенциями:

Таблица 1

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

Виды деятельности	Профессионально-специализированные компетенции, установленные самостоятельно или трудовые функции (формируются и (или) совершенствуются) ПСК	Практический опыт	Знания	Умения
1	2	3	4	5
<p>Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта</p>	<p>ПСК-1¹. Разработка процедур интеграции программных модулей</p>	<p>- выявлять проблемы сопряжения неоднородных модулей и компонентов компьютерного программного обеспечения;</p> <p>оптимизировать программный код с использованием специализированных программных средств;</p>	<p>- методов и средств сборки модулей и компонентов компьютерного программного обеспечения;</p> <p>- интерфейсов взаимодействия с внешней средой;</p> <p>- интерфейсов взаимодействия внутренних модулей системы;</p> <p>- методов и средств разработки процедур для развертывания компьютерного программного обеспечения</p> <p>- методов и средств миграции и преобразования данных</p> <p>- языков, утилит и среды программирования, средств пакетного выполнения процедур</p> <p>- основных стандартов оформления технической документации на компьютерное</p>	<p>- писать программный код процедур интеграции программных модулей;</p> <p>- использовать выбранную среду программирования для разработки процедур интеграции программных модулей;</p> <p>- применять методы и средства сборки модулей и компонентов компьютерного программного обеспечения, разработки процедур для развертывания компьютерного программного обеспечения, миграции и преобразования данных, создания программных интерфейсов;</p> <p>- применять заданные стандарты и шаблоны для составления и оформления технической</p>

¹ Профессиональный стандарт 06.001 Программист, утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. N 424н, обобщенная трудовая функция С, трудовая функция С/01.5

			программное обеспечение языки, утилиты и среды программирования, средств пакетного выполнения процедур;	документации; - осуществлять коммуникации с заинтересованным и сторонами
Разработка и отладка программного кода	ПСК-2² . Проверка и отладка программного кода	<ul style="list-style-type: none"> - анализировать и проверять исходный программный код; - отладить программный код на уровне программных модулей; - отладить программный код на уровне межмодульных взаимодействий и взаимодействий с окружением; - оценивать и согласовывать сроки выполнения поставленных задач. 	<ul style="list-style-type: none"> - методов и приема отладки программного кода; - типа и формата сообщений об ошибках, предупреждений; - способов использования технологических журналов, форматы и типы записей журналов; - современных компиляторов, отладчика и оптимизаторов программного кода; - сообщений о состоянии аппаратных средств 	<ul style="list-style-type: none"> - выявлять ошибки в программном коде; - применять методы и приемы отладки программного кода; - интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов; - применять современные компиляторы, отладчики и оптимизаторы программного кода; - осуществлять коммуникации с заинтересованным и сторонами.
Универсальные компетенции (УК)³				
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.	<ul style="list-style-type: none"> - владеть методами поиска, сбора и обработки, критического анализа и синтеза информации; - владеть методикой системного подхода для решения 	<ul style="list-style-type: none"> - методики поиска, сбора и обработки информации; - актуальных российских и зарубежных источников информации в сфере профессиональной деятельности; 	<ul style="list-style-type: none"> - анализировать задачу, выделять её базовые составляющие, осуществлять декомпозицию задачи; - находить и критически анализировать информацию, необходимую для решения 	УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.

² Профессиональный стандарт 06.001 Программист, утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. N 424н, обобщенная трудовая функция А, трудовая функция А/05.3

³ Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата) (Приказ Министерства науки и высшего образования РФ от 19 сентября 2017 г. N 922)

	поставленных задач.	- метода системного анализа.	поставленной задачи; - рассматривать возможные варианты решения задачи, оценивая их достоинства и недостатки; - грамотно, логично, аргументированно формировать собственные суждения и оценки.	
--	---------------------	------------------------------	--	--

1.4. Категория слушателей

Данная программа предназначена для граждан, для граждан, заинтересованных в получении дополнительных компетенций в области информационных технологий, а именно — в разработке программных продуктов на языке Python и подходящих под одну из категорий, утверждённых Министерством труда и социальной защиты в рамках нацпроекта «Кадры».

Требования к слушателям программы: высшее или среднее профессиональное образование, подтвержденное документами государственного образца.

1.5. Формы и технологии обучения

Форма обучения – очно-заочная с применением электронного обучения (ЭО) и дистанционных образовательных технологий (ДОТ).

1.6. Период обучения, срок освоения и режим занятий

Период обучения – 5 недель.

Общая трудоемкость программы 72 академических часа: часов контактной работы со слушателем – 54, часов самостоятельной работы – 16, часов итоговой аттестации – 2.

Режим занятий: 4-6 дней с понедельника по субботу с 09.00 до 16.00.

1.7. Документ о квалификации

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

2. СОДЕРЖАНИЕ ПРОГРАММЫ

Таблица 2

Календарный учебный график

1 неделя	2 неделя	3 неделя	4 неделя	5 неделя .
<i>УЗ ДОТ, СР ЭО, ТКУ ДОТ</i>	<i>УЗ ДОТ, СР ЭО, ТКУ ДОТ, ИА ДОТ</i>			

Календарный учебный график содержит следующие условные обозначения:

УЗ ДОТ – учебные занятия с применением дистанционных образовательных технологий;

ТКУ ДОТ – текущий контроль успеваемости с применением дистанционных образовательных технологий;

ИА ДОТ – итоговая аттестация с применением дистанционных образовательных технологий;

СР ЭО – самостоятельная работа с применением электронного обучения.

2.3. Содержание программы повышения квалификации

Таблица 4

Содержание программы по темам

Номер темы (раздела)	Содержание темы (раздела)
1. Основы Python. Базовые структуры.	<p>Понятие языка в программировании. Синтаксис и семантика. Знакомство с языком программирования Python. Установка Python на компьютер. Работа с интерпретатором языка. Среда разработки IDLE. Интерфейс редактора кода VS Code. Среда разработки PyCharm. Пакетные менеджеры. Miniconda, pip. Блокноты Jupyter Notebook. Комерческие сервисы.</p> <p>Знакомство с языком программирования Python. Особенности. Работа в интерпретаторе языка. Переменные. Типы данных: int, bool, NoneType, string, list. Методы и встроенные функции. Операции над типами данных. Файл и компиляция кода.</p> <p>Строки и операции над строками. Операции над строками. Дополнительные возможности функции print. Ввод значений с клавиатуры. Строковые методы.</p> <p>Коллекции: Списки, кортежи. Сравнение списков и кортежей. Последовательность исполнения команд. Операторы. Условный оператор if. Цикл for и while. Применение и особенности операторов. Динамическая и статическая типизация. Кортежи tuple и распаковка переменных. Словари и множества. Применение типов данных. Исключения. Конструкция switch-case.</p> <p>Применение функций. Параметры функций. Типы параметров. Правила вызова функции. Строки документации.</p>
2. Импорт. Работа с файлами.	<p>Область видимости переменных. Модули Python. Импорт. Варианты импорта. Формирование модуля. Пакеты. Импорт из пакетов. Отладчик. Как работает Python. Отличие от других языков.</p> <p>Стандартная библиотека Python. random, os, time, sys, pprint, decimal и тд. Модуль re. Работа с файлами.</p> <p>Работа с файлами. Функция open(). Кодировки, декодирование. Байтовые строки. Оператор with. Работа с json. Работа с xlsx. Работа с xml. Работа с csv. Сбор данных с помощью рекурсии.</p>
3. Применение алгоритмов для решения задач.	<p>Алгоритмизация: алгоритмы и способы их описания, составление алгоритмов на языке блок-схем, базовые управляющие конструкции алгоритмов.</p> <p>Бинарный поиск. Варианты сортировки. Сортировка выбором. Сортировка пузырьком. Сортировка расческой. Сортировка вставками. Быстрая сортировка. Сортировка подсчетом. Пирамидальная сортировка. Сортировка слиянием. Рекурсия подробнее. Хеш-таблицы. Реализация алгоритмов на Python. Сравнение скорости алгоритмов.</p>
4. Прикладные решения для поддержки кода.	<p>Системы контроля версий SCM. Какие бывают и для чего используются. Git, как необходимый инструмент для передачи кода. Принципы работы систем контроля версий. Отличительные особенности Git. Введение в Git. Установка и начальная настройка Git. Создание, изменение и удаление репозитория.</p>

	<p>Принципы ветвления и слияния. Управление ветками. Решение конфликтов. Коммит. Логирование. Откат назад.</p> <p>Репозитории. Хостинги. Работа в mos.hub. Регистрация, создание репозитория. Подключение локального устройства к репозиторию. Документация проекта README.md. Разметка Markdown. Файл requirements.txt. Лицензии. Значки. GitHub.</p>
<p>5. Объектно-ориентированное программирование на Python.</p>	<p>Основы объектно-ориентированного подхода в программировании. Понятие объекта в программировании. Объект в Python. Императивный стиль программирования, отличия от объектно-ориентированного подхода в программировании. Понятие класса в Python. Поток программы. Методы и атрибуты. Пространство имен класса. Параметры методов. Параметр self. Связанный метод. Вызов атрибутов и методов. Экземпляр класса.</p> <p>Специальные (магические) методы Python. Конструктор <code>__init__</code>. Деструктор <code>__del__</code>. Метод <code>__str__</code>. Имя экземпляра. Метод <code>__len__</code>. Метод <code>__bool__</code>. Перегрузка операторов. Переопределение методов. Метод <code>__eq__</code>. Метод <code>__hash__</code>. Метод <code>__add__</code>. Метод <code>__call__</code>.</p> <p>Наследование классов. Атрибут <code>__class__</code>. Иерархия классов. Принципы ООП. Инкапсуляция. Метод <code>super()</code>. Множественное наследование.</p>
<p>6. Функциональное программирование на Python.</p>	<p>Функциональный стиль в программировании. Функции высшего порядка. Функции в качестве параметров. Имена функции. Функции <code>filter</code>, <code>map</code>. Объявление функции внутри другой функции. Однострочные функции <code>lambda</code>.</p> <p>Списковые и словарные сборки. Генерация функций. Функции генераторы. Функции итераторы. Функции декораторы. Рецепторы.</p>
<p>7. Работа с базами данных.</p>	<p>Структура базы данных. Функции SQL. Достоинства SQL. Формы использования SQL. Программирование запросов выборки данных.</p> <p>Реляционные базы данных. Модели БД и схемы. Требования к БД. Первичный и внешний ключи. SQL синтаксис. Sqlite3. Запросы SQL. Команда SHOW. Синтаксис запроса SELECT. Запросы к одной таблице. Многотабличные и вложенные запросы. Оптимизация запросов на выборку данных инструментов. Объединение таблиц JOIN.</p> <p>Работа с внешними пакетами. Интеграция Python и SQL. Работа с СУБД. Создание, изменение и удаление базовых таблиц. Команды CREATE, DELETE, UPDATE, INSERT.</p>

8. Отраслевое применение языка Python.

Промышленное программирование. Поддержка цикла разработки. Отладка кода. Введение в тестирование. Тесты и примеры.
Введение в логирование. Логирование в несколько мест назначения. Конфигурирование логирования. Рефакторинг.
Параллельные вычисления. Асинхронное программирование. Потоки на классах. Проблемы гонки состояний для потоков. Очереди для обмена данными между потоками. Многопроцессорное программирование. Блокировка потоков.
Этапы создания проекта. Настройка окружения. Индексатор пакетов `pip`. Фреймворки. `Telebot`. Взаимодействие скрипта `python` и телеграм. Администрирование бота. Проект бота на базе `aiogram`. Структура бота. Выделение классов. Запуск.

3. ОРГАНИЗАЦИОННЫЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Материально-техническое и программное обеспечение реализации программы

РАНХиГС располагает материально-технической базой, обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работы, итоговой аттестации слушателей, предусмотренных учебным планом вуза и соответствующей действующим санитарным и противопожарным правилам и нормам.

Минимально необходимый для реализации программы перечень материально-технического обеспечения включает в себя:

✓ лекционные аудитории (оборудованные видеопроекторным оборудованием для презентаций, средствами звуковоспроизведения, экраном и имеющие выход в Интернет);

✓ помещения для проведения семинарских и практических занятий (оборудованные учебной мебелью);

✓ компьютерные классы с персональными компьютерами с программным обеспечением: интерпретатор языка программирования Python

Программное обеспечение: лицензионные системные программы операционные системы (Windows, Python, иные), обеспечивающие взаимодействие всех других программ с оборудованием и взаимодействие пользователя персонального компьютера с программами. Универсальные офисные прикладные программы и средства ИКТ, например, программа подготовки презентаций; использование Интернет, электронной почты; использование автоматизированных поисковых систем Интернет.

Структура информационно-образовательной среды включает:

- образовательный w-портал (сайт) Академии;
- базы данных электронных информационных ресурсов;
- корпоративную сеть Академии.

W- портал (сайт) обеспечивает через Интернет:

- доступ к электронным информационным и образовательным ресурсам Академии;

- доступ к нормативным и организационно-методическим документам, регламентирующим образовательный процесс в Академии;

- систему электронной почтовой пересылки письменных работ слушателей;

- взаимодействие слушателей с преподавателями, организаторами образовательного процесса и администрацией Академии.

Корпоративная сеть обеспечивает коммуникацию преподавателей, сотрудников и слушателей. Занятия с применением ДОТ проводятся на электронной площадке mts-link.ru и др., для технического сопровождения проведения занятий с применением ДОТ привлекаются IT-специалисты.

Организация учебного процесса по программе основана на технологиях, позволяющих повысить эффективность получаемого образования.

Слушатели программы на период обучения обеспечиваются раздаточными материалами (учебно-методическими пособиями, информационными и справочными материалами) по тематике занятий, раздаточными материалами на бесплатной, безвозмездной основе. Также участники программы получают доступ к информационным ресурсам (источникам) и средствам информатизации.

3.2 Учебно-методическое и информационное обеспечение программы

Все слушатели обеспечиваются учебно-методическими пособиями и рекомендациями в печатном или электронном виде. Каждый слушатель обеспечивается доступом к электронно-библиотечной системе сети РАНХиГС при Президенте РФ.

Глубокое усвоение материала обеспечивается сочетанием аудиторных занятий и самостоятельной работы слушателей с литературой и нормативными документами. Основным видом учебных занятий по данной дисциплине являются лекции.

В рамках самостоятельной работы слушатели разбирают вопросы по теме занятий и осуществляют подготовку к зачету. Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы слушателей. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов.

Примеры заданий для самостоятельных работ и практических занятий:

Задание 1

Согласно вашему варианту, выполните следующие пункты:

1. Изучить описание схемы базы данных;
2. Опишите таблицы с помощью сущностей и атрибутов;
3. Определите первичные ключи и прочие свойства;
4. Распределите связи между сущностями согласно описанию.

Сценарий:

Университет ведет учет информации о студентах, их научных руководителях, а также о факультетах, на которых они находятся.

Таблицы:

Преподаватели:

- ID преподавателя
- Звание
- Имя преподавателя
- Фамилия преподавателя

Факультет:

- ID факультета
- ID студента
- ID преподавателя
- Аудитория
- Наименование

Студенты:

- ID студента
- Имя студента
- Фамилия студента
- Курс

Связи:

- Каждый преподаватель прикреплен к одному факультету. На факультете может быть много преподавателей.
- Каждый студент закреплен за одним факультетом. На факультете может быть много студентов.

Задание 2

1. Ознакомиться с презентациями
2. Установить на компьютер Git
 - a. Для Windows установить git и открыть git bash
 - b. Для MacOS открыть терминал
 - i. Вставить следующую команду `/bin/bash . -c "$(curl -fsSLhttps://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
 - ii. затем ввести `brew install git`
3. В папке “Документы” создать папку `project_01`
4. В git bash или терминале введите команду `cd /Documents/project_01`.
Так откроется папка проекта в терминале.
5. Задайте email и имя с помощью команды.
 - a. `git config --global user.name "ваш никнейм"`
 - b. `git config --global user.email "ваш email"`
6. Инициализируйте проект с помощью команды `git init`
7. Добавьте два файла (`page_01.txt`, `page_02.txt`) в папку `project_01`, скачав их по ссылкам.

8. Добавьте файлы в список отслеживаемых с помощью *git add*. (вместе сточкой!!!)
9. Сделайте коммит с помощью команды *git commit -m "page_01, page_02 added"*
10. Для создания репозитория на GitHub ознакомьтесь с инструкцией в видео “Как создать репозиторий GitHub (ДЗ)”

Нормативно-правовые документы:

1. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных».
2. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации».
3. Приказ ФСТЭК России от 18.02.2013 № 21 «Об утверждении Составы и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».
4. Приказ ФСТЭК России от 11.02.2013 № 17 «Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах».
5. Постановление Правительства РФ от 01.11.2012 № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных».
6. Стратегия развития отрасли информационных технологий в Российской Федерации на 2025-2035 годы (утверждается Правительством РФ).

Основная литература:

1. Черпаков, И. В. Алгоритмизация и программирование на Python : учебник для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2026. — 159 с. — (Высшее образование). — ISBN 978-5-534-21910-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582412>.
2. Воган Л. «Непрактичный» Python: занимательные проекты для тех, кто хочет поумнеть: Пер. с англ. — СПб.: БХВ-Петербург, 2021. — 464 с.
3. Майер К. Однострочники Python: лаконичный и содержательный код. — СПб.: Питер, 2022. — 256 с.: ил. — (Серия «Библиотека программиста»). — ISBN 978-5-4461-2966-9.
4. Федоров, Д. Ю. Программирование на Python : учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Высшее образование). —

ISBN 978-5-534-19666-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/556864>.

Дополнительная литература:

1. Шоттс У. Командная строка Linux. Полное руководство. — 2-е изд. — СПб.: Питер, 2025. — 512 с.: ил. — (Серия «Для профессионалов»). // Обновленное издание классического руководства.
2. Kubernetes in Action, Second Edition. By Marko Luksa. Manning Publications, 2026. // Актуальное руководство по оркестрации контейнеров с учетом всех последних изменений в экосистеме Kubernetes.
3. Джонс К. Быстрый старт с ИИ для разработчиков: интеграция GPT, Gemini и других моделей в ваши проекты. — М.: ДМК Пресс, 2025. — 320 с. // Практическое руководство по использованию современных языковых моделей в разработке.
4. Свифт Д. Осознанный Python: паттерны, производительность и архитектура современных приложений. — Пер. с англ. — СПб.: Питер, 2026. — 432 с. — (Серия «Библиотека программиста»). // Книга о продвинутых аспектах Python для создания надежных и масштабируемых систем.
5. Гарсия, М. и Прайс, Б. Git для профессионалов: углубленное понимание системы контроля версий. — O'Reilly Media, 2025. — 310 р. // Продвинутое руководство, заменяющее и дополняющее базовые материалы по Git, с фокусом на сложные сценарии и внутреннее устройство.
6. Архитектура данных в эпоху AI: принципы и практики. / Под ред. С. Власовой. — СПб.: БХВ-Петербург, 2026. — 408 с. // Сборник статей о проектировании систем данных, готовых к интеграции с машинным обучением и аналитикой.
7. Сикорски Р. Rust в enterprise: создание безопасных и высокопроизводительных сервисов. — Addison-Wesley Professional, 2026. // Фокус на применении Rust для backend-разработки в корпоративной среде.

Интернет-ресурсы:

1. <https://pythonworld.ru/> - сайт по языку программирования Python
2. <https://metanit.com/python/tutorial/> - Руководство по языку программирования Python
3. <https://practicum.yandex.ru/ycloud/> - Онлайн-курс «Инженер облачных сервисов». Yandex Cloud. Яндекс Практикум.
4. <https://www.coursera.org/specializations/python> - Онлайн-курс

- «Специализация Python для всех». University of Michigan. Coursera.
5. <https://code-basics.com/ru/languages/python> - Бесплатный курс по Python для начинающих
 6. <https://ru.hexlet.io/blog/posts/programmirovaniye-na-python-osobennosti-obucheniya-perspektivy-situatsiya-na-rynke-truda> - Программирование на Python: особенности обучения, перспективы, ситуация на рынке труда

4. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Для текущего контроля успеваемости слушателя используются фронтальный опрос, а также тестирование по теоретическим и практическим аспектам.

Примеры вопросов для текущего контроля:

1. Какие секции включает в себя запрос SELECT?
2. Что может представлять собой возвращаемый элемент в запросе SELECT?
3. Каким образом может быть задано условие поиска применительно к однотабличным запросам?
4. Какие виды функций используются в СУБД?
5. Опишите особенности синтаксиса и работы нескольких функций каждого вида.
6. Как отсортировать результаты запроса на выборку данных?
7. Как сформировать запрос с группировкой? Какие ограничения накладываются на такие запросы?
8. Каким образом используется секция HAVING в запросе SELECT?
9. В чем состоит отличие в работе секций HAVING и WHERE?
10. Что такое вложенный запрос?

Примеры тестов для текущего контроля

1. Каким будет результат выполнения кода?
`print (9.0//2)`

- а) 4.5
- б) 4.0
- в) 4
- г) Error

2. Каким будет результат выполнения кода?
`varOne = 1`
`varTwo = 2`
`print(varOne and varTwo)`

- а) 1
- б) 2
- в) True
- г) False

3. Каким будет результат выполнения кода?
`fnum = (2 * 4) % 3`
`snum = 3 * 2 % 3`

```
tnum = 5 * (1 % 2)
print('fnum', fnum, 'snum', snum, 'tnum', tnum)
```

- а) fnum 0 snum 0 tnum 5
- б) fnum 2 snum 1 tnum 5
- в) fnum 2 snum 0 tnum 5
- г) Error

4. Каким будет результат выполнения кода?

```
var = '11_22'
fvar = float(var)
print(fvar)
```

- а) 11.22
- б) 1122
- в) 1122.0
- г) Error

5. Каким будет результат выполнения кода?

```
var = 3 / 2
print(var == 1,5)
```

- а) True
- б) False
- в) True 5
- г) False 5
- д) Error

6. Какой синтаксис корректен для возвращения первого символа строки?

```
x = 'Я люблю Python'
```

- а) y1 = x[0]
- б) y2 = x.sub(0, 1)
- в) y3 = sub(x, 0, 1)

7. Каким будет результат выполнения кода?

```
svar = "Ещё один тест :("
svar[2] = "e"
print(svar)
```

- а) Ещё один тест :(
- б) Ещёё один тест :(
- в) Ещё один тест :(

г) Error

8. Каким будет результат выполнения кода?

```
lst = [13, 56, 17]
lst.append([87])
lst.extend([45, 67])
print(lst)
```

- а) [13, 56, 17, 87, 45, 67]
- б) [13, 56, 17, [87], 45, 67]
- в) [13, 56, 17, [87], [45, 67]]
- г) [13, 56, 17, 87, [45, 67]]

9. Каким будет результат выполнения кода?

```
a = True
b = False
c = False
```

```
if not a or b:
    print(1)
elif not a or not b and c:
    print(2)
elif not a or not b and not b and a:
    print(3)
else:
    print(4)
```

- а) 1
- б) 2
- в) 3
- г) 4
- д) Error

10. Каким будет результат выполнения кода?

```
dct = {'one': 1, 'two': 2, 'three': 3}
while dct:
    print(dct.popitem(), end = '<->')
print('done')
```

- а) done
- б) one<->two<->three
- в) 1<->2<->3
- г) ('one', 1)<->('two', 2)<->('three', 3)<->done
- д) ('three', 3)<->('two', 2)<->('one', 1)<->done

Примеры практических заданий для текущего контроля.

Упражнение 1

В високосный год к февралю добавляется один день - 29 февраля. Чтобы определить, является ли год високосным, существуют два правила:

- Год, номер которого кратен 4 (делится на 4 без остатка), то год является високосным (например, 2020, 2024).
- Год НЕ является високосным, если его номер кратен 100 без остатка (например, 1900). Из этого правила исключаются те случаи, когда год одновременно кратен и 100, и 400 - в этом случае он високосный (например, 2000).

Напишите программу, в которую пользователь вводит год, а программа выводит сообщение, является ли год високосным или нет. Примеры работы программы:

```
C:\python>py app.py
Введите год: 2024
Год високосный
```

```
C:\python>py app.py
Введите год: 1900
Год не високосный
```

Упражнение 2

Напишите программу, в которую пользователь вводит два числа, а программа выводит сообщение, какое из чисел наибольшее. Примеры работы программы:

```
Введите число a: 5
Введите число b: 6
Наибольшее число: 6
```

Упражнение 3

Измените код предыдущей программы, чтобы она вычисляла наибольшее число с помощью одного выражения.

Решение

Упражнение 4

Напишите программу, которая вычисляет сумму скидки в зависимости от суммы продажи. Пусть скидки установлены следующим образом:

Сумма продажи	Скидка
0-5000	5%
5000-15000	12%

15000-25000	20%
свыше 25000	30%

После вычисления скидки программа должна вывести саму скидку и сумму с учетом скидки. Например:

Введите сумму продажи: 25000
 Скидка: 5000.0
 Сумма с учетом скидки : 20000.0

Итоговая форма контроля по программе:

Итоговая аттестация проводится в форме зачета и включает:

- Тестирование на знание теоретических аспектов;
- Задачу на проверку практических навыков работы

Пример задачи:

Расширенная индексация

1. Создайте новый проект в VS Code. Скопируйте условие задачи и необходимые переменные ниже.

Есть словарь кодов товаров

```
titles = {
    'Кроссовки тип 3 - Adidas': '100000110',
    'Мячик тип 2 - Adidas': '100000146',
    'Кепка тип 1 - Adidas': '100000149',
    'Ремень тип 2 - Nike': '100000194',
    'Футболка тип 1 - Adidas': '100000224',
    'Шапка тип 5 - Puma': '100000280',
}
```

Есть словарь списков словарей количества товаров на складе.

```
sales = {
    '100000110': [{'quantity': 31, 'price': 1637}],
    '100000146': [ {'quantity': 4, 'price': 45}, {'quantity': 10, 'price': 48}],
    '100000149': [ {'quantity': 28, 'price': 279}, {'quantity': 32, 'price': 291}],
    '100000194': [{'quantity': 8, 'price': 220}, {'quantity': 1, 'price': 170}],
    '100000224': [{'quantity': 61, 'price': 438}, {'quantity': 23, 'price': 302},
    {'quantity': 50, 'price': 412}],
    '100000280': [{'quantity': 26, 'price': 175}, ]
}
```

Рассчитать на какую сумму лежит каждого товара на складе.

#

```

# Вывести суммарную стоимость каждого товара на складе с помощью
циклов
# Формат строки вывода: "<товар> - <кол-во> шт, стоимость <общая
стоимость> руб"
#
# Псевдок алгоритма
# цикл for по товарам с получением кода и названия товара
# инициализация переменных для подсчета количества и стоимости
товара
# получение списка на складе по коду товара
# цикл for по списку на складе
# подсчет количества товара
# подсчет стоимости товара
# вывод на консоль количества и стоимости товара на складе

```

2. После завершения работы сделайте push на ваш репозиторий GitHub. Пришлите ссылку на выполненную работу преподавателю.

Пример тестового задания:

1. Какие характеристики можно отнести к языку программирования Python?
 - а) интерпретируемый
 - б) с динамической типизацией
 - в) использующий препроцессор для макроподстановок
 - г) для быстрой разработки приложений
2. Какие характеристики можно отнести к языку программирования Python?
 - а) имеет эффективный компилятор в коды процессора
 - б) использует раннее связывание
 - в) объектно-ориентированный
 - г) универсальный язык программирования
3. Какие характеристики можно отнести к языку программирования Python?
 - а) для быстрой разработки приложений
 - б) богатый и громоздкий синтаксис
 - в) поощряет повторное использование кода
4. Какие парадигмы и стили программирования поддерживает Python ?
 - а) объектно-ориентированный
 - б) модульный
 - в) программирование в ограничениях
 - г) функциональное программирование
5. Что будет выведено следующей программой:


```
a = 1
```



```
for i in range(1, 10):
```

```
    if i % 2 == 0:
```

```
        S = S + i
```

```
print S
```

а) 0

б) 10

в) 20

г) 30

11. Как на языке Python записывается логическая операция "И" (умножение) x на y?

а) x and y

б) x in y

в) x + y

г) y or x

12. s – строка. Что будет вычислено при вызове программой функции len(s)?

а) Сумма элементов строки

б) Элемент строки s

в) Длина строки

г) Функцию len(s) нельзя вычислить

13. Что произойдет при исполнении следующего кода?

```
a=input('Введите значение переменной')
```

а) На экран будет выведено значение переменной a

ПК -6
б) На экран будет выведено сообщение «Введите значение переменной»

и после

ввода значения от пользователя оно будет присвоено переменной a// +
отмечен

один правильный ответ

в) Код будет исполнен с ошибкой

г) Переменной a строкового типа будет присвоено значение «Введите значение

переменной»

14. Какой тип или структуру данных получаем на выходе после операции divmod()?

а) Строка

б) Кортеж

в) Число

г) Список

15. Что будет выведено на экране в результате выполнения кода:

```
print(['domik']*3)
```

а) ['domik'], ['domik'], ['domik']

б) [['domik']] в) ['domik'], ['domik'], ['domik']

г) Нет правильного ответа

16. Какой код нужно использовать, чтобы открыть файл `c:\scores.txt` для чтения?
- а) `infile = open("c:\scores.txt", "r")`
 - б) `infile = open("c:\\scores.txt", "r")`
 - в) `infile = open(file = "c:\scores.txt", "r")`
 - г) `infile = open(file = "c:\\scores.txt", "r")`
17. Какой будет результат `int(12.5)`?
- а) 12
 - б) 12.5
 - в) 12,5
 - г) 13
18. Числа в Python могут быть:
- а) целыми
 - б) действительными
 - в) вещественными
19. Изменяемыми последовательностями в Python являются:
- а) строки
 - б) кортежи
 - в) списки
20. Что будет выведено на экран после выполнения этой программы:
- ```
s = "123"
s = s + s[1] + s + s[2]
print s
```
- а) 12321233
  - б) 12311232
  - в) 12321232
  - г) 12311233

### **Критерии оценки:**

Часть 1. Критерии выставления оценок за тест, состоящий из 20 заданий.

Время выполнения работы: 20 мин.

«Зачтено» – 8-20 правильных ответов;

«Не зачтено» – менее 8 правильных ответов.

Часть 2 оценивается за логичность, практическую применимость рекомендаций и корректность расчетов.

Критерии оценивания практического задания.

«Зачтено» выставляется, если слушатель показал знание учебного материала по теме практической работы, показывает усвоение взаимосвязи основных понятий используемых в работе, смог ответить на все уточняющие и дополнительные вопросы.

«Не зачтено» выставляется слушателю, если он имеет существенные пробелы в знаниях основного учебного материала практической работы, который полностью не раскрыл содержание вопросов, не смог ответить на уточняющие и дополнительные вопросы.

## 5. ИНДИКАТОРЫ СФОРМИРОВАННЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКА ПРОГРАММЫ

В результате освоения программы у слушателя сформированы компетенции:

Таблица 5

| Компетенция (код, содержание)                                                                                                         | Индикаторы                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ПСК-1. Разработка процедур интеграции программных модулей                                                                             | <p>Применяет базовые информационные технологии при решении прикладных задач.</p> <p>Решает задачи по сбору, анализу и визуализации данных с помощью прикладных программных продуктов (в т.ч. на языке Python).</p>                                                                                                                                                                                        |
| ПСК-2. Проверка и отладка программного кода                                                                                           | <p>Применяет Python программирования для написания программного кода;</p> <p>Использует выбранную среду программирования и средства системы управления базами данных;</p> <p>Оптимизирует программный код с использованием специализированных программных средств;</p>                                                                                                                                    |
| УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач. | <p>Знает базовые принципы сбора, отбора и обобщения информации в целях проведения и построения логических и математических моделей поставленных задач.</p> <p>Умеет выделять данные, которые необходимо собирать для построения логических и математических моделей поставленных задач, проводить их первичную обработку</p> <p>Имеет практический опыт поиска источников информации по заданной теме</p> |

## ВНУТРЕННЯЯ РЕЦЕНЗИЯ

на дополнительную профессиональную программу повышения квалификации «Python-разработчик»

**Категория слушателей программы:** лица, имеющие среднее профессиональное и (или) высшее образование, специалисты, стремящиеся к приобретению компетенций в области программирования на языке Python, веб-разработки и анализа данных.

**Актуальность программы:** Актуальность программы продиктована стремительным ростом спроса на Python-разработчиков в сферах веб-разработки, анализа данных и искусственного интеллекта, а также необходимостью цифровой трансформации бизнеса, требующей специалистов, способных эффективно работать с данными и писать качественный программный код.

**Цель программы:** подготовка востребованных специалистов в области программирования на языке Python.

**Основное направление подготовки:** изучение основ синтаксиса языка, типов данных и функций, объектно-ориентированного программирования, работы с файлами и базами данных, основ веб-разработки на Flask/Django, а также введение в анализ данных с библиотеками Pandas и NumPy.

**Особенности программы:** программа выстроена по принципу «от простого к сложному» — от консольных приложений к веб-проектам. Сильной стороной является преподавательский состав, имеющий многолетний практический опыт работы разработчика в IT-компаниях реального сектора экономики. Программа реализуется с использованием дистанционных образовательных технологий (ДОТ), что позволяет вовлекаться в учебный процесс представителям разных регионов России.

**Срок реализации программы:** указывается в соответствии с учебным планом (72 часа).

**Заключение:**

Рецензируемая программа повышения квалификации «Python-разработчик», разработанная факультетом менеджмента Института управления РАНХиГС, отличается новизной в подаче материала и актуальностью содержания, способствует подготовке конкурентоспособных IT-специалистов для современной цифровой экономики. Программа соответствует требованиям, предъявляемым к дополнительным профессиональным программам повышения квалификации, и рекомендуется к реализации.

Рецензент \_\_\_\_\_



С.В. Коробейникова

к.э.н., доцент, профессор кафедры  
менеджмента факультета

менеджмента Института управления

## РЕЦЕНЗИЯ

на программу повышения квалификации «Python-разработчик», разработанную факультетом менеджмента Института управления РАНХиГС

Рецензируемая дополнительная профессиональная программа (ДПП) повышения квалификации «Python-разработчик» разработана на факультете менеджмента ФГБОУ ВО «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» в соответствии с требованиями рынка IT-труда и тенденциями развития языков программирования.

Целью реализации программы является формирование у слушателей профессиональных компетенций, необходимых для разработки программного обеспечения на языке Python. Общая характеристика программы содержит информацию о цели, планируемых результатах (владение синтаксисом, библиотеками, фреймворками), категории слушателей (лица с базовыми навыками алгоритмизации) и форме обучения.

Содержание учебной программы включает календарный учебный график, учебный план и рабочие программы модулей. В разделе организационно-педагогического обеспечения представлены кадровое, материально-техническое, программное и учебно-методическое обеспечение. Учебный план состоит из тематических блоков, охватывающих ключевые аспекты разработки: от основ синтаксиса до создания веб-приложений и работы с базами данных через SQLAlchemy.

Качество содержательной составляющей учебного плана не вызывает сомнений. Включенные в план модули раскрывают сущность современных подходов к программированию. Структура плана логична и последовательна.

В учебном процессе прослеживается использование активных и интерактивных форм проведения занятий, включая хакатоны, разбор кода, парное программирование. Оценка качества освоения программы включает текущий контроль (ТКУ) в виде проверки кода, промежуточную аттестацию (ПА) в виде выполнения практических заданий и итоговую аттестацию (ИА) в форме защиты индивидуального проекта.

В целом рецензируемая программа отвечает основным требованиям, способствует качественной подготовке Python-разработчиков и рекомендована для реализации на факультете менеджмента Института управления РАНХиГС.

Рецензент: Главный научный сотрудник ЦЭМИ РАН, д.э.н., доцент

Жданов Д.А.

Дата:

Подпись:



**Сведения о преподавательском составе и ведущих специалистах  
(кадровая справка)**

| Ф.И.О.<br>преподавателя<br>/ведущего<br>специалиста | Квалификация по<br>диплому о высшем<br>или среднем<br>профессиональном<br>образовании                                                                                                       | Дополнительн/ая/ые<br>квалификаци/я/и | Ученая<br>степень,<br>ученое<br>(почетное)<br>звание | Стаж работы в<br>области<br>профессионально<br>й деятельности | Наименование<br>преподаваемой<br>дисциплины/темы<br>(модуля)/стажиров<br>ки (при наличии)                                                                                                                                                                                                               |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                                                   | 2                                                                                                                                                                                           | 3                                     | 4                                                    | 5                                                             | 6                                                                                                                                                                                                                                                                                                       |
| Пышков<br>Никита<br>Игоревич                        | Российская академия народного хозяйства и государственной службы при Президенте РФ; магистр по направлению «38.04.02. Менеджмент» по специальности «Управление производственными системами» | -                                     | -                                                    | 8                                                             | Основы Python. Базовые структуры. Импорт. Работа с файлами. Применение алгоритмов для решения задач. Прикладные решения для поддержки кода. Объектно-ориентированное программирование на Python. Функциональное программирование на Python. Работа с базами данных. Отраслевое применение языка Python. |