

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский Томский государственный университет»



УТВЕРЖДАЮ

Проректор по образовательной
деятельности

Е.В. Луков

(подпись)

«23» июня 2023 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Python Essentials: концепции и технологии разработки»

технической направленности

Уровень сложности: «Продвинутый»

144 часа

СОГЛАСОВАНО:

Директор ЦРСК

Заместитель директора ИДО

М.В. Назарова

В.С. Дубровская

г. Томск - 2023




Пояснительная записка

<p>Актуальность программы</p>	<p>Изучение Python Essentials: концепции и технологии разработки является актуальным и востребованным в настоящее время по нескольким причинам:</p> <ul style="list-style-type: none"> ● Python является одним из наиболее популярных языков программирования в мире. Он используется для решения различных задач, таких как веб-разработка, научные вычисления, машинное обучение, анализ данных и многих других. ● Python обладает простым и понятным синтаксисом, что делает его доступным для изучения даже для начинающих программистов. ● Python имеет большое сообщество разработчиков и пользователей, что обеспечивает широкую поддержку и возможность получения помощи в случае необходимости. ● Python является основным языком программирования для многих известных фреймворков и библиотек, таких как Django, Flask, NumPy, Pandas, TensorFlow и многих других. ● Python является языком программирования, который может быть использован в различных областях, включая научные исследования, инженерное дело, финансы, медицину и т.д. ● Python Essentials: концепции и технологии разработки предоставляет необходимые знания и навыки для работы с языком Python, что может привести к повышению квалификации и расширению карьерных возможностей. <p>Таким образом, изучение Python Essentials: концепции и технологии разработки является актуальным и востребованным в настоящее время и может быть полезным как для начинающих программистов, так и для опытных разработчиков.</p>
<p>Цель программы</p>	<p>Цель программы "Python Essentials: концепции и технологии разработки" заключается в том, чтобы обучить участников основным концепциям и технологиям языка программирования Python, который является одним из наиболее популярных языков программирования в мире. Программа предназначена для тех, кто только начинает свой путь в программировании, а также для тех, кто уже имеет некоторый опыт в этой области.</p>
<p>Планируемые результаты обучения (знания, умения, навыки)</p>	<p>Сформировать у обучающихся школ базовые знания и умения языка программирования Python, Сформировать умение применять программные средства для решения задач из различных предметных областей способствует развитию логического и комбинаторного мышления</p>
<p>Категория обучающихся по программе</p>	<p>Учащиеся 8 класса Учащиеся 9 класса</p>

	Учащиеся 10 класса Учащиеся 11 класса Обучающиеся по программам среднего профессионального образования
Срок освоения программы	2 года
Формы и режим занятий	Очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения
Форма обучения	Очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения
Трудоемкость программы	144 ак.ч.
Примечание	Продвинутый уровень

Об организации

Наименование поля	Допустимые значения поля	Значение поля
ИНН организации, осуществляющей образовательную деятельность	10 арабских цифр	7018012970
Наименование организации	строка	Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет»
Логотип организации	изображение в формате jpeg разрешением не менее 100x100 пиксель	 Национальный исследовательский Томский государственный университет
Ссылка на логотип организации	URL на изображение, находящееся в сети интернет	https://drive.google.com/file/d/170I9BfWy1szAoJPBAKlaWFIgwrlJiqYT/view?usp=share link
Контакты ответственного за программу. ФИО	строка от 5 до 255 символов	Шарыпина Полина Андреевна
Контакты ответственного за программу. Должность	строка от 5 до 255 символов	Заместитель директора Центра совместных образовательных программ
Контакты ответственного за программу. Телефон	Формат +7(XXX)XXXXXX X	89528957687
Контакты ответственного за программу. E-mail	строка	sharypinapolina@gmail.com
Название программы (курса)	строка	Python Essentials: концепции и технологии разработки

<p>Описание программы</p>	<p>строка не менее 1000 не более 5000 символов</p>	<p>Дополнительная общеобразовательная программа «Python Essentials: концепции и технологии разработки» является неотъемлемой частью образовательной программы ТГУ и дает возможность каждому ребенку получать дополнительное образование исходя из его интересов, склонностей и способностей, образовательных потребностей, осуществляемых за пределами федеральных государственных образовательных стандартов и федеральных государственных требований. По своему функциональному назначению программа является общеразвивающей и направлена на удовлетворение потребностей, обучающихся в интеллектуальном, нравственном совершенствовании, в организации их свободного времени. Направленность общеобразовательной программы «Python Essentials: концепции и технологии разработки» техническая.</p>
<p>Аннотация</p>	<p>строка до 1000 символов</p>	<p>Программа «Python Essentials: концепции и технологии разработки» направлена на предоставление учащимся фундаментальных знаний и навыков в программировании на языке Python. В рамках курса учащиеся изучают основные концепции языка, включая переменные, типы данных, управляющие конструкции, функции и модули. Кроме того, они познакомятся с различными инструментами и технологиями, используемыми в разработке на языке Python, такими как NumPy, Pandas, Matplotlib, Flask и др.</p> <p>Требования к ученикам (уровень знаний):</p> <ul style="list-style-type: none"> ● Базовые навыки владения компьютером (запуск приложений, создание папок, смена имени файла, скачивание картинок из интернета). ● Начальные знания английского языка и умение бегло печатать на английской раскладке.

		<p>Результат обучения:</p> <ul style="list-style-type: none"> ● Уверенное владение языком Python и его основными конструкциями, такими как условные операторы, циклы и функции. ● Понимание принципов программирования игр и создание собственных проектов на Python ● Навыки работы с графическим интерфейсом пользователя (GUI) и создание собственных приложений с помощью библиотеки Tkinter. ● Умение создавать и настраивать графические элементы интерфейса, такие как кнопки, поля ввода и метки. ● Навыки обработки пользовательского ввода и отображения результатов на экране. ● Понимание принципов организации и структурирования кода, а также методологии разработки программных проектов. ● Уверенность в работе с командной строкой и управлении файлами в операционной системе.
<p>Цель программы</p>	<p>строка не менее 100 символов</p>	<p>Цель программы "Python Essentials: концепции и технологии разработки" заключается в том, чтобы обучить участников основным концепциям и технологиям языка программирования Python, который является одним из наиболее популярных языков программирования в мире. Программа предназначена для тех, кто только начинает свой путь в программировании, а также для тех, кто уже имеет некоторый опыт в этой области.</p>
<p>Актуальность</p>	<p>строка не менее 500 символов</p>	<p>Изучение Python Essentials: концепции и технологии разработки является актуальным и востребованным в настоящее время по нескольким причинам:</p> <ul style="list-style-type: none"> ● Python является одним из наиболее популярных языков программирования в мире. Он используется для решения различных

задач, таких как веб-разработка, научные вычисления, машинное обучение, анализ данных и многих других.

- Python обладает простым и понятным синтаксисом, что делает его доступным для изучения даже для начинающих программистов.
- Python имеет большое сообщество разработчиков и пользователей, что обеспечивает широкую поддержку и возможность получения помощи в случае необходимости.
- Python является основным языком программирования для многих известных фреймворков и библиотек, таких как Django, Flask, NumPy, Pandas, TensorFlow и многих других.
- Python является языком программирования, который может быть использован в различных областях, включая научные исследования, инженерное дело, финансы, медицину и т.д.
- **Python Essentials: концепции и технологии разработки**

предоставляет необходимые знания и навыки для работы с языком Python, что может привести к повышению квалификации и расширению карьерных возможностей.

Таким образом, изучение **Python Essentials: концепции и технологии разработки** является актуальным и востребованным в настоящее время и может быть полезным как для начинающих программистов, так и для опытных разработчиков.

Дополнительная информация	строка	
Формат обучения	значение из: "Онлайн"	Онлайн
	"Оффлайн" "Смешанный"	
Уровень сложности	значение из: "Начальный" " "Базовый" "Продвинутый"	Продвинутый
Срок освоения образовательной программы	строка	144 часа
Объем каждого модуля в ак.ч.	целое число	36
Объем часов в неделю в ак.ч.	целое число	4
Минимальное количество человек на одном потоке курса	целое число <100000	24
Количество уроков	целое число	116
Данные о количестве школьников, ранее успешно прошедших обучение по образовательной программе	целое число, при наличии	0
Направленность программы	строка	Современные языки программирования
Язык программирования	строка	Язык программирования Python
Дополнительная общеобразовательная программа не представлена для участия в иных федеральных проектах, направленных на дополнительное образование граждан, кроме федерального	строка, значения: "Представлена"/ "Не представлена"	Не представлена

проекта «Развитие кадрового потенциала ИТ-отрасли»		
Образовательная программа не была реализована до начала отбора и/или не реализовывается в период отбора на безвозмездной основе	строка, значения "Реализована ранее"/ "Не реализована"	Не реализована
Дополнительная общеобразовательная программа не была реализована до начала отбора и/или не реализовывается в период отбора на безвозмездной основе	строка, значение: "Первая версия образовательной программы, ранее не реализованная" "Ранее реализованная версия образовательной программы"	Ранее не реализованная версия образовательной программы
Категория обучающихся (возраст) по программе	строка, значение: "Учащиеся 8 класса" "Учащиеся 9 класса" "Учащиеся 10 класса" "Учащиеся 11 класса"	Учащиеся 8 класса Учащиеся 9 класса Учащиеся 10 класса Учащиеся 11 класса Обучающиеся по программам среднего профессионального образования
Описание планируемых результатов обучения	строка	Сформировать у обучающихся школ базовые знания и умения языка программирования Python, Сформировать умение применять программные средства для решения задач из различных предметных областей способствует развитию логического и комбинаторного мышления
Ссылка на лендинг	строка	https://it-school.tgu-dpo.ru/python-essentials

Ссылка на LMS	строка	https://odin.study/ru/
Страница обучения на курсе	строка	https://www.odin.study/ru/EducationalProgram/Info/7392
Итоговая аттестация. Количество академических часов	Целое число больше 0	2
Итоговая аттестация. Формы контроля	строка от 4 до 255 символов	Зачет (Итоговое практическое задание)
Итоговая аттестация. Диагностические инструменты	строка не менее 10 символов	<p>При оценке надо руководствоваться следующими критериями:</p> <ol style="list-style-type: none"> 1) полнота и правильность при создании проекта; 2) степень осознанности, понимания изученного; 3) оформление ответа. <p>Оценка «5» ставится, если:</p> <ol style="list-style-type: none"> 1) проект по теме «Python Essentials: концепции и технологии разработки» выполнен без ошибок, полно изложен изученный материал; 2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные; 3) в чек-листе излагает материал последовательно и правильно (в соответствии с шаблоном) с точки зрения преподавателя. <p>Оценка «4» ставится, если ученик выполнил проект, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности оформления излагаемого материала в чек-листе.</p> <p>Оценка «3» ставится, если ученик освоил знания и понимает основы проект по теме «Python Essentials: концепции и технологии разработки», но:</p>

		<p>1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;</p> <p>2) не может обосновать свои суждения и привести свои примеры;</p> <p>3) допускает ошибки в оформлении чек-листа.</p> <p>Оценка «2» ставится, если ученик обнаруживает незнание большей части</p> <p>изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал.</p> <p>Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овла</p> <p>Итоговый контроль проходит в конце курса .</p> <p>Формы проведения аттестации: зачетная работа.</p> <p>В таблице представлен перечень заданий, которые необходимо выполнить в период обучения на курсе.</p>
Итоговая аттестация. Показатели и критерии оценивания	строка не менее 50 символов	<p>Для допуска к итоговой аттестации слушателю необходимо набрать не менее 3 баллов за выполнение каждого практического задания из разделов курса.</p> <p>- итоговая работа, включающая разработку проекта в соответствии с установленными требованиями. Итоговая аттестация проходит в конце обучения в асинхронном формате.</p>
Итоговая аттестация. Шкала оценивания, нижнее значение	целое число	2
Итоговая аттестация. Шкала оценивания, верхнее значение	целое число, большее чем нижнее значение шкалы	5

Итоговая аттестация. Шкала оценивания, минимальный проходной балл	целое число в диапазоне шкалы	3
--	-------------------------------	----------

Аттестация

Промежуточная аттестация		
Количество академических часов	строка не менее 10 символов	4 часа текущая работа, включающая выполнение 4 практических задания
Формы контроля	строка не менее 10 символов	Для допуска к итоговой аттестации слушателю необходимо набрать не менее 3 баллов за выполнение каждого практического задания из разделов курса.
Диагностические инструменты	строка не менее 10 символов	Оценка полноты выполнения работы. Оценка способности оперировать полученными знаниями и умениями при решении практической задачи. Проверка результата запуска программного решения на языке Python.
Показатели и критерии оценивания	строка не менее 10 символов	Шкала оценивания: Нижнее значение 2 Верхнее значение 5 Минимальный проходной балл для успешной сдачи 3 При оценке надо руководствоваться следующими критериями: 1) полнота и правильность выполнения заданий; 2) степень осознанности, понимания изученного; 3) оформление ответа.

		<p>Оценка «5» ставится, если:</p> <p>1) Практическое задание выполнено без ошибок, полно изложен изученный материал;</p> <p>2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные;</p> <p>Оценка «4» ставится, если ученик выполнил практическое задание, удовлетворяющее тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности излагаемого материала.</p> <p>Оценка «3» ставится, если ученик освоил знания и понимает основы практического задания, но:</p> <p>1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;</p> <p>2) не может обосновать свои суждения и привести свои примеры;</p> <p>3) допускает ошибки в оформлении.</p> <p>Оценка «2» ставится, если ученик обнаруживает незнание большей части</p> <p>изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал.</p> <p>Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.</p>
<p>Примеры заданий</p>	<p>строка не менее 10 символов</p>	<p>Модуль 1. Практическое задание № 1. Основы языка Python</p> <p>Шаг 1: Установка Python</p> <p>Первым шагом необходимо установить интерпретатор языка Python на ваш компьютер. Это можно сделать, загрузив установочный файл Python с официального сайта Python</p>

(<https://www.python.org/downloads/>) и следуя инструкциям по установке.

Шаг 2: Напишите программу на Python, которая запрашивает у пользователя имя и возраст, а затем выводит сообщение, которое содержит это имя и возраст. Например, "Привет, Маша! Тебе 25 лет." и

Шаг 3: Напишите программу на Python, которая находит сумму всех чисел от 1 до 1000.

Шаг 4: Напишите программу на Python, которая принимает на вход два числа и выводит наибольшее из них.

Шаг 5: Напишите программу на Python, которая запрашивает у пользователя строку и затем выводит ее в обратном порядке. Например, "Привет" должно стать "тевирП".

Шаг 7: Напишите программу на Python, которая проверяет, является ли заданное число простым числом.

Шаг 8. Сохраните файлы и выложите их в Odin

Модуль 2. Практическое задание № 2. Работа с данными в Python

Шаг 1. Открыть среду разработки Idle

Шаг 2. Создать новый проект

Шаг 3. Напишите программу на Python, которая считывает данные из файла "input.txt", вычисляет сумму всех чисел в этом файле и записывает результат в файл "output.txt".

Шаг 4. Пример содержимого файла "input.txt":

10

20

30

Пример содержимого файла "output.txt":

60

Шаг 5. Программа считывает каждую строку из файла "input.txt" и преобразует ее в целое число. Затем она вычисляет сумму всех чисел в списке и записывает результат в файл "output.txt".

Шаг 6. Сохраните файл и выложите его в Odin

Модуль 3. Практическое задание № 3. Работа с библиотеками Python

Шаг 1. Открыть среду разработки Idle

Шаг 2. Создать новый проект

Шаг 3. Импортируйте библиотеку NumPy.

Шаг 4. Создайте случайный массив чисел с помощью функции `np.random.rand()`, передав ей желаемый размер массива в качестве аргумента. Например, можно создать массив размером 5x5

Шаг 5. Выведите созданный массив на экран.

Шаг 6. Отсортируйте созданный массив по возрастанию с помощью функции `np.sort()`.

Шаг 7. Выведите отсортированный массив на экран.

Шаг 8. Отсортируйте созданный массив по убыванию с помощью функции `np.sort()` и передав ей аргумент -1.

Шаг 9. Выведите отсортированный по убыванию массив на экран.

Шаг 10. Проверьте, что созданный массив не изменился, выведя его на экран еще раз.

Шаг 11. Завершите программу.\

Шаг 12. Сохраните файл и выложите его в Odin/

Модуль 4. Практическое задание № 4. Разработка приложений на Python

Шаг 1. Открыть среду разработки Idle

Шаг 2. Создать новый проект

Шаг 3: Создайте новый файл с расширением `.py` и назовите его "sum.py"

Шаг 4: Напишите код, который просит пользователя ввести первое число и сохраняет его в переменную. Используйте функцию `input ()` для запроса ввода от пользователя.

Шаг 5: Напишите код, который просит пользователя ввести второе число и сохраняет его в переменную. Используйте функцию `input()` для запроса ввода от пользователя.

Шаг 6: Преобразуйте две переменные, которые вы получили в шагах 2 и 3, в числа. Используйте функцию `int()` для преобразования строки в целое число.

		<p>Шаг 7: Напишите код, который складывает два числа, которые вы получили в шагах 4, и сохраняет результат в новую переменную.</p> <p>Шаг 8: Напишите код, который выводит результат сложения на экран. Используйте функцию <code>print()</code> для вывода результата.</p> <p>Шаг 9: Запустите приложение и убедитесь, что оно работает правильно. Если необходимо, исправьте ошибки в вашем коде.</p> <p>Шаг 10: Дополнительное задание. Измените приложение таким образом, чтобы оно принимало не только целые числа, но и числа с плавающей точкой. Для этого используйте функцию <code>float()</code> вместо <code>int()</code> в шаге 4.</p> <p>Шаг 11. Сохраните файл и выложите его в Odin</p>
Шкала оценивания, нижнее значение	строка не менее 10 символов	Шкала оценивания: Нижнее значение 2
Шкала оценивания, верхнее значение	строка не менее 10 символов	Шкала оценивания: Верхнее значение 5
Шкала оценивания, минимальный проходной балл	строка не менее 10 символов	Шкала оценивания: Минимальный проходной балл для успешной сдачи 3

I ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «**Python Essentials: концепции и технологии разработки**» является неотъемлемой частью образовательной программы Федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет» и дает возможность каждому ребенку получать дополнительное образование исходя из его интересов, склонностей и способностей, образовательных потребностей, осуществляемых за пределами федеральных государственных образовательных стандартов и федеральных государственных требований.

По своему функциональному назначению программа является общеразвивающей и направлена на удовлетворение потребностей, обучающихся в интеллектуальном, нравственном совершенствовании, в организации их свободного времени.

Направленность общеобразовательной программы «**Python Essentials: концепции и технологии разработки**» техническая.

Актуальность и педагогическая целесообразность программы

Python является одним из наиболее популярных языков программирования, который используется в различных областях, включая разработку игр и создание графических пользовательских интерфейсов. Python Essentials предоставляет базовые знания и навыки по языку программирования Python, который является одним из наиболее популярных языков в мире. Python используется в различных областях, таких как научные вычисления, машинное обучение, веб-разработка, анализ данных, автоматизация и многих других. Изучение Python позволяет получить необходимые знания и навыки для работы в этих областях. Python Essentials представляет собой комплексный курс, который охватывает различные аспекты языка программирования Python. Курс структурирован таким образом, чтобы начинающие пользователи могли изучать язык с самых основ, а опытные пользователи могли повысить свой уровень. Программа включает в себя практические задания и проекты, которые помогают закрепить полученные знания. Python Essentials разработан таким образом, чтобы обучение проходило поэтапно и последовательно, с учетом уровня подготовки учащихся. Курс включает в себя дополнительные материалы, такие как видеолекции, упражнения, тесты и многие другие, которые помогают закрепить теоретические знания на практике. Кроме того, курс также предоставляет возможность получения обратной связи и помощи от опытных преподавателей и наставников. В настоящее время рынок труда требует высококвалифицированных специалистов, которые могут работать с языком программирования Python. Python Essentials может быть полезным как для учащихся, которые только начинают свой путь в области программирования, так и для опытных разработчиков, которые хотят расширить свои знания и навыки.

Цель и основные задачи программы: формирование первичных компетенций в области информационных технологий, создания игр, навыков исследовательской и изобретательской деятельности, проектной и командной работы.

Основные задачи:

Образовательные:

- Ознакомление с основами языка Python: курс включает в себя изучение синтаксиса языка Python, работу с переменными, операторами и структурами данных.
- Предоставление практических навыков: Python Essentials включает в себя множество практических заданий, которые помогают закрепить теоретические знания на практике. Учащиеся изучают различные аспекты языка Python, такие как функции, модули, обработка исключений, работа с файлами и многие другие.
- Обучение работе с библиотеками: Python Essentials включает в себя обучение работе с различными библиотеками языка Python, такими как NumPy, Pandas и Matplotlib. Эти библиотеки широко используются в области анализа данных и машинного обучения.
- Развитие навыков программирования: Python Essentials помогает развить навыки программирования, такие как создание и отладка кода, работа с Git и GitHub, написание тестов и многие другие.
- Подготовка к дополнительным курсам: Python Essentials может служить основой для дополнительных курсов по Python, таких как курсы по машинному обучению, веб-разработке и другим.

Воспитательные:

- воспитать мотивацию учащихся к изобретательству, созданию собственных игр;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- формировать правильное восприятие системы ценностей, принципов, правил информационного общества;
- формировать потребность в самостоятельном приобретении и применении знаний, потребность к постоянному саморазвитию;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

Развивающие:

- способствовать развитию творческих способностей учащихся, познавательных интересов, развитию индивидуальности и самореализации;
- расширять технологические навыки при подготовке различных информационных материалов;
- развивать познавательные способности ребенка, память, внимание, пространственное мышление, аккуратность и изобретательность при выполнении учебных проектов;
- формировать творческий подход к поставленной задаче;
- развивать навыки инженерного мышления, умения работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;
- формировать навыки рефлексивной деятельности.

Педагогические принципы, определяющие теоретические подходы к построению образовательного процесса

Реализация программы «Python Essentials: концепции и технологии разработки» основывается на принципах научности, последовательности, системности, связи теории с практикой, доступности.

В целях раскрытия педагогического и развивающего потенциала учебно-воспитательного процесса по программе акцент в ней делается на следующих принципах:

Принцип прочности предполагает стремление к тому, чтобы приобретенные знания, умения и навыки обучающихся были прочно закреплены, в первую очередь, практически.

Для этого необходимо их участие в соревнованиях, где полученные умения и навыки можно проверить на практике, в выездных мероприятиях, а также постоянное поддержание интереса обучающихся к занятиям.

Принцип проектности предполагает последовательную ориентацию всей деятельности педагога на подготовку и выведение ребенка в самостоятельное проектное действие, развертываемое в логике замысел – реализация – рефлексия. В ходе проектирования перед человеком всегда стоит задача представить себе еще не существующее, но то, что он хочет, чтобы появилось в результате его активности.

Принцип активности предполагает необходимость строить учебный процесс таким образом, чтобы учащиеся не просто воспринимали предлагаемый материал, но и стремились закрепить полученные знания, умения и навыки, анализировали ошибки и достижения свои и товарищей, самостоятельно решали поставленные задачи.

Основные характеристики образовательного процесса

Уровень освоения программы базовый- продвинутый, что предполагает освоение обучающимися специализированных знаний, обеспечение трансляции общей и целостной картины тематического содержания программы.

Форма обучения: очная форма с применением дистанционных образовательных технологий.

Срок реализации программы - 2 года.

Режим работы: занятия проводятся 2-3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

Продолжительность образовательного процесса: начало занятий 01 октября 2023 г., завершение 31 мая 2024 г.

Объем учебных часов по программе 144 часа.

Ожидаемые результаты освоения программы

Личностные:

- развита способность переносить (выдерживать) известные нагрузки в течение определенного времени, преодолевать трудности;
- умеет контролировать свои поступки;
- умеет воспринимать общие дела как свои собственные.

Метапредметные:

- развито умение представлять результаты своей работы окружающим, аргументировать свою позицию;
- развита способность правильно организовывать рабочее место и время для достижения поставленных целей.

- сформировано у обучающихся умение работать в команде.

Предметные:

- сформированы практические и теоретические знания в области устройства и функционирования современных платформ;
- изучены основы алгоритмизации, построения алгоритмов;
- научился формулировать и анализировать алгоритмы;
- изучить основы программирования на языке Python
- научиться разрабатывать игры на Python с помощью библиотеки Pygame и Tkinter.

Данная программа адаптирована для занятий с учащимися среднего и старшего возраста, с различной степенью подготовки, включает теоретические и практические задания.

Входные требования к обучающимся: тестирование, базовое владение персональным компьютером.

Для занятий -наличие ПК, камеры, микрофона, стабильного доступа в интернет.

Календарный учебный график формируется непосредственно при реализации программы общеобразовательной программы «**Python Essentials: концепции и технологии разработки**».

Первый модуль с 01.10.2023 - 30.11.2023

Второй модуль: с 01.12 2023 - 31.01.2024

Третий модуль: с 01.02.2024 - 31.03.2024

Четвертый модуль: с 01.04.2024 - 31.05.2024

Занятия проводятся 2-3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

Преподаватели

<p>ФИО</p>	<p>Наименование основного места работы</p>	<p>Должность</p>	<p>Высшее образование или среднее профессиональное образование по направлению «Образование и педагогические науки»</p>	<p>Высшее образование или среднее профессиональное образование по иному направлению соответствующим направлениям ДОП</p>	<p>Ссылка на веб-страницы с портфолио</p>	<p>Информация о курсах повышения квалификации или профессионального преподавателя дисциплины (за последние 3 года)</p>	<p>Пройдена промежуточная аттестация не менее чем за два года обучения по образовательным программам высшего образования по специально-стям и направлениям подготовки, соответствующим ющим направлениям ДОП</p>	<p>Отметка о получении согласия на обработку персональных данных</p>
-------------------	---	-------------------------	---	---	--	---	---	---

строка от 2 до 100 СИМВОЛОВ	строка от 2 до 255 СИМВОЛОВ.	строка от 2 до 255 СИМВОЛОВ	да/нет	да/нет	строка	да/нет	да/нет
Сергеев Олег Владимирович	ГБОУ Школа имени В.В.Маяковского	Учитель информатики и робототехники	да	да	<u>Сергеев О.В.</u> (резюме 2021).pdf - Яндекс.Документы (yandex.ru)	Разработка веб-приложений с использованием JavaScript ФГАОУ ВО "Национальный исследовательский университет "Высшая школа экономики". Московский институт электроники и математики им. А. Н. Тихонова 2019 год.	да
							да

Максимов Егор Васильевич	GeekBrains Fogstream Robotex Центр развития роботехники	Преподаватель	да	да	да	<p>Навыки: C# Python Django Framework Git HTML5 CSS3 JavaScript SQL Adobe Photoshop Blender 3D Bootstrap Администрирование серверов Linux PostgreSQL C++ MySQL Roblox Lua Linux HTML CSS jQuery SOLID ООП MVC Docker Базы данных Разработка ПО Agile Nginx</p>	да
Пьянков Михаил Андреевич	Unity и Unreal Engine	Преподаватель	да	да	да	<p>Telegram- и Discord-боты: -</p>	да

						<p>Электронный помощник, - Бот, сообщаящий погоду через API.</p> <p>Интернет-магазин с помощью Python Flask и шаблонизатор а Jinja с подключение м БД и библиотек Selenium, BeautifulSoup Soup.</p> <p>Аудио-плеер и фото-редактор с помощью Tkinter.</p> <p>Анализ данных на Python ML с помощью библиотек Pandas, NumPy, Matplotlib, Seaborn</p>		
--	--	--	--	--	--	--	--	--

СОДЕРЖАНИЕ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

**СОДЕРЖАНИЕ УЧЕБНОГО (ТЕМАТИЧЕСКОГО) ПЛАНА
«Python Essentials: концепции и технологии разработки»**

№	п/п	Наименование модулей (курсов)	Общая трудоемкость, ч.	Всего контактных часов			Контактные часы		СРС, ч.	Формы контроля
				Общее количество Лекций и ПЗ	ДЗ	Лекции	Практические занятия			
I		Модуль 1: Основы языка Python	36	30	1	9	21	6	Зачет	
1.1		Тема 1.1 Введение в Python	1	1		1				
1.2		Тема 1.2 Переменные и типы данных	1	1		1				

1.3	Тема 1.3 Управляющие конструкции	1	1	1	1	1	1	1					
1.4	Тема 1.4 Функции	7	6					1	5	1			
1.5	Тема 1.5 Строки и последовательности	7	6					1	5	1			
1.6	Тема 1.6 Списки и кортежи	4	3					1	2	1			
1.7	Тема 1.7 Словари и множества	5	4					1	3	1			
1.8	Тема 1.8 Обработка ошибок и отладка	5	4					1	3	1			
1.9	Тема 1.9 Модули и пакеты	5	4				1	1	3	1			Зачет
2	Модуль 2: Работа с данными в Python	36	30	1	9	21	6	Зачет					
2.1	Тема 2.1 Основы работы с данными	2	2					1	1				
2.2	Тема 2.2 Чтение и запись данных	3	3					1	2				
2.3	Тема 2.3 Работа с базами данных	4	3					1	2	1			
2.4	Тема 2.4 Обработка и анализ данных	4	3					1	2	1			
2.5	Тема 2.5 Визуализация данных	4	3					1	2	1			
2.6	Тема 2.6 Работа с API	4	4					1	3				
2.7	Тема 2.7 Web-скрапинг	3	3					1	2				
2.8	Тема 2.8 Регулярные выражения	4	3					1	2	1			
2.9	Тема 2.9 Работа с XML и JSON	4	3				1	1	2	1			
2.10	Тема 2.10 Работа с CSV и Excel	4	3						3	1			Зачет
3	Модуль 3: Работа с библиотеками Python	36	30	1	10	20	6	Зачет					

3.1	Тема 3.1 NumPy	4	3		1	2	1	
3.2	Тема 3.2 Pandas	4	3		1	2	1	
3.3	Тема 3.3 Matplotlib	4	3		1	2	1	
3.4	Тема 3.4 PyTorch	4	3	1	1	2	1	Зачет
3.5	Тема 3.5 Seaborn	3	3		1	2		
3.6	Тема 3.6 Scikit-learn	4	4		1	3		
3.7	Тема 3.7 TensorFlow	4	4		1	3		
3.8	Тема 3.8 Beautiful Soup	4	3		1	2	1	
3.9	Тема 3.9 Django	5	4		2	2	1	
4	Модуль 4: Разработка приложений на Python	34	28	1	9	19	6	Зачет
4.1	Тема 4.1 Основы разработки на Python	5	4		1	3	1	
4.2	Тема 4.2 Создание консольных приложений	4	4		2	2		
4.3	Тема 4.3 Разработка графических интерфейсов	5	4		1	3	1	
4.4	Тема 4.4 Разработка веб-приложений	4	3		1	2	1	
4.5	Тема 4.5 Разработка игр на Python	5	4		1	3	1	
4.6	Тема 4.6 Разработка приложений для мобильных устройств	6	5		2	3	1	
4.7	Тема 4.7 Создание API на Python	5	4	1	1	3	1	Зачет

									Итоговая проектная работа
5.	Итоговая аггестация	2		2					2
	Итого	144	120	6	37	83	24		

II. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ (формы аттестации, оценочные и методические материалы)

Условия прохождения курса

В курсе будут оцениваться:

текущая работа, включающая выполнение 4 практических задания.

Задания будут оцениваться по пятибалльной шкале :

Шкала оценивания:

Нижнее значение 2

Верхнее значение 5

Минимальный проходной балл для успешной сдачи 3

При оценке надо руководствоваться следующими критериями:

- 1) полнота и правильность выполнения заданий;
- 2) степень осознанности, понимания изученного;
- 3) оформление ответа.

Оценка «5» ставится, если:

- 1) Практическое задание выполнено без ошибок, полно изложен изученный материал;
- 2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные;

Оценка «4» ставится, если ученик выполнил практическое задание, удовлетворяющее тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности излагаемого материала.

Оценка «3» ставится, если ученик освоил знания и понимает основы практического задания, но:

- 1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;
- 2) не может обосновать свои суждения и привести свои примеры;
- 3) допускает ошибки в оформлении.

Оценка «2» ставится, если ученик обнаруживает незнание большей части изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал.

Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.

Для допуска к итоговой аттестации слушателю необходимо набрать не менее 3 баллов за выполнение практических заданий из разделов курса (50% заданий , должно быть оценено минимум на 3 балла, каждое).

- итоговая работа, включающая разработку проекта в соответствии с установленными

требованиями. Итоговая аттестация проходит в конце обучения в асинхронном формате. Максимальное количество баллов, которое слушатели могут получить за итоговую аттестацию – 5

Для успешного завершения обучения по курсу необходимо, чтобы 50 % заданий в рамках текущей работы были зачтены (оценены преподавателем на 3, 4 или 5 баллов).

Итоговая аттестационная работа была зачтена (оценены преподавателем на 3, 4 или 5 баллов).

В таблице представлен перечень заданий, которые будут проводиться на практике в режиме онлайн с преподавателем в период обучения на курсе.

№ п/п	Мероприятия промежуточного контроля	Количество баллов
1.	Практическое задание № 1. Основы языка Python	2-5
2.	Практическое задание № 2. Работа с данными в Python	2-5
3.	Практическое задание № 3. Работа с библиотеками Python	2-5
4.	Практическое задание № 4. Разработка приложений на Python	2-5

В таблице представлен перечень заданий, которые необходимо выполнить в период обучения на курсе.

В качестве итоговой работы слушателям предлагается разработать проект по теме «Python Essentials: концепции и технологии разработки»

Проект представляет собой Создание игры "Пинг-понг" с помощью Python с GUI и Pygame. Подготовить чек лист по проделанной работе в документе word. Расписать в свободной форме.

При оценке надо руководствоваться следующими критериями:

- 1) полнота и правильность при создании проекта;
- 2) степень осознанности, понимания изученного;
- 3) оформление ответа.

Оценка «5» ставится, если:

- 1) проект по теме «Python Essentials: концепции и технологии разработки» выполнено без ошибок, полно изложен изученный материал;
- 2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные;

3) в чек-листе излагает материал последовательно и правильно (в соответствии с шаблоном) с точки зрения преподавателя.

Оценка «4» ставится, если ученик выполнил проект, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности оформления излагаемого материала в чек-листе.

Оценка «3» ставится, если ученик освоил знания и понимает основы проект по теме «**Python Essentials: концепции и технологии разработки**», но:

- 1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;
- 2) не может обосновать свои суждения и привести свои примеры;
- 3) допускает ошибки в оформлении чек-листа.

Оценка «2» ставится, если ученик обнаруживает незнание большей части изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал.

Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овла

Итоговый контроль проходит в конце курса .

Формы проведения аттестации: зачетная работа.

В таблице представлен перечень заданий, которые необходимо выполнить в период обучения на курсе.

Итоговый контроль проходит в конце курса .

Формы проведения аттестации: зачетная работа.

Оценочные материалы для текущего контроля успеваемости:

Модуль 1. Практическое задание № 1. Основы языка Python

Шаг 1: Установка Python

Первым шагом необходимо установить интерпретатор языка Python на ваш компьютер. Это можно сделать, загрузив установочный файл Python с официального сайта Python (<https://www.python.org/downloads/>) и следуя инструкциям по установке.

Шаг 2: Напишите программу на Python, которая запрашивает у пользователя имя и возраст, а затем выводит сообщение, которое содержит это имя и возраст. Например, "Привет, Маша! Тебе 25 лет." и

Шаг 3: Напишите программу на Python, которая находит сумму всех чисел от 1 до 1000.

Шаг 4: Напишите программу на Python, которая принимает на вход два числа и выводит наибольшее из них.

Шаг 5: Напишите программу на Python, которая запрашивает у пользователя строку и затем выводит ее в обратном порядке. Например, "Привет" должно стать "тевирП".

Шаг 7: Напишите программу на Python, которая проверяет, является ли заданное число простым числом.

Шаг 8. Сохраните файлы и выложите их в Odin

Модуль 2. Практическое задание № 2. Работа с данными в Python

- Шаг 1. Открыть среду разработки Idle
Шаг 2. Создать новый проект
Шаг 3. Напишите программу на Python, которая считывает данные из файла "input.txt", вычисляет сумму всех чисел в этом файле и записывает результат в файл "output.txt".
Шаг 4. Пример содержимого файла "input.txt":
10
20
30
Пример содержимого файла "output.txt":
60
Шаг 5. Программа считывает каждую строку из файла "input.txt" и преобразует ее в целое число. Затем она вычисляет сумму всех чисел в списке и записывает результат в файл "output.txt".
Шаг 6. Сохраните файл и выложите его в Odin

Модуль 3. Практическое задание № 3. Работа с библиотеками Python

- Шаг 1. Открыть среду разработки Idle
Шаг 2. Создать новый проект
Шаг 3. Импортируйте библиотеку NumPy.
Шаг 4. Создайте случайный массив чисел с помощью функции `np.random.rand()`, передав ей желаемый размер массива в качестве аргумента. Например, можно создать массив размером 5x5
Шаг 5. Выведите созданный массив на экран.
Шаг 6. Отсортируйте созданный массив по возрастанию с помощью функции `np.sort()`.
Шаг 7. Выведите отсортированный массив на экран.
Шаг 8. Отсортируйте созданный массив по убыванию с помощью функции `np.sort()` и передав ей аргумент -1.
Шаг 9. Выведите отсортированный по убыванию массив на экран.
Шаг 10. Проверьте, что созданный массив не изменился, выведя его на экран еще раз.
Шаг 11. Завершите программу.
Шаг 12. Сохраните файл и выложите его в Odin/

Модуль 4. Практическое задание № 4. Разработка приложений на Python

- Шаг 1. Открыть среду разработки Idle
Шаг 2. Создать новый проект
Шаг 3: Создайте новый файл с расширением .py и назовите его "sum.py"
Шаг 4: Напишите код, который просит пользователя ввести первое число и сохраняет его в переменную. Используйте функцию `input()` для запроса ввода от пользователя.
Шаг 5: Напишите код, который просит пользователя ввести второе число и сохраняет его в переменную. Используйте функцию `input()` для запроса ввода от пользователя.
Шаг 6: Преобразуйте две переменные, которые вы получили в шагах 2 и 3, в числа. Используйте функцию `int()` для преобразования строки в целое число.
Шаг 7: Напишите код, который складывает два числа, которые вы получили в шагах 4, и сохраняет результат в новую переменную.
Шаг 8: Напишите код, который выводит результат сложения на экран. Используйте функцию `print()` для вывода результата.
Шаг 9: Запустите приложение и убедитесь, что оно работает правильно. Если необходимо, исправьте ошибки в вашем коде.
Шаг 10: Дополнительное задание. Измените приложение таким образом, чтобы оно принимало не только целые числа, но и числа с плавающей точкой. Для этого используйте функцию `float()` вместо `int()` в шаге 4.

Шаг 11. Сохраните файл и выложите его в Odin

Условия допуска к итоговой аттестации

Для допуска к итоговой аттестации слушатели должны выполнить все задания и получить не менее 3 баллов за каждое из них.

Итоговая аттестация

В качестве итоговой работы слушателям предлагается разработать проект по теме «**Python Essentials: концепции и технологии разработки**»

Проект представляет собой создание программы, которая будет анализировать текст и определять, какие слова встречаются в нем наиболее часто. Сделать чек лист по проделанной работе в документе word.

Расписать в свободной форме.

Максимальное количество баллов, которое слушатели могут получить за итоговую аттестацию – 5 (минимальное количество баллов за итоговую аттестацию – 3).

Задание

Итоговое задание по теме **Python Essentials: концепции и технологии разработки** будет заключаться в создании программы, которая будет анализировать текст и определять, какие слова встречаются в нем наиболее часто.

Этапы работы

Шаг 1. Напишите программу, которая будет считывать текст из файла. Для этого можно использовать функцию `open()` и метод `read()`.

Шаг 2. Приведите текст к нижнему регистру и удалите из него все знаки препинания. Для этого можно использовать методы `lower()` и `translate()`.

Шаг 3. Разделите текст на слова. Для этого можно использовать метод `split()`.

Шаг 4. Создайте словарь, в котором ключами будут слова, а значениями — количество их вхождений в текст. Для этого можно использовать цикл `for` и методы `get()` и `count()`.

Шаг 5. Отсортируйте словарь по убыванию значений. Для этого можно использовать функцию `sorted()` и передать ей параметр `key=lambda x: x[1], reverse=True`.

Шаг 6. Выведите на экран первые 10 самых часто встречающихся слов.

Итог работы

Мы открываем файл 'text.txt' и считываем из него текст. Затем мы приводим текст к нижнему регистру и удаляем знаки препинания.

Далее мы разделяем текст на слова и создаем словарь, в котором ключами являются слова, а значениями — количество их вхождений в текст. Затем мы сортируем словарь по убыванию значений и выводим первые 10 самых часто встречающихся слов.

На данном зачете выдается набор задач, которые, ученик должен решить по мере усвоения программы. Данные задачи представлены в таблице 1.1.

III. КАДРОВЫЕ УСЛОВИЯ

Руководитель программы:

Сергеев Олег Владимирович, Российский государственный социальный университет, Москва Факультет: Информационные технологии (бакалавриат) Специальность: Педагогическое образование (информатика) Сентябрь 2011 - Июль 2015 Российский государственный социальный университет, Москва Факультет: Информационные технологии (магистратура) Специальность: Педагогическое образование (информатика) Сентябрь 2016 - Июль 2019

Стаж работы – более 7 лет.

Сертификаты/ курсы:

Разработка веб-приложений с использованием Javascript ФГАОУ ВО "Национальный исследовательский университет "Высшая школа экономики". Московский институт электроники и математики им. А. Н. Тихонова 2019 год.

Программирование веб-сайтов и веб-приложений на HTML и CSS ФГАОУ ВО "Национальный исследовательский университет "Высшая школа экономики". Московский институт электроники и математики им. А. Н. Тихонова 2018 год.

Методика и использование 3d-прототипирования и моделирования на учебных занятиях в инженерном классе Академия «Просвящение» 2017 год.

«Подготовка технических специалистов, оказывающих информационно-техническую помощь руководителю и организаторам пункта проведения экзамена при проведении государственной итоговой аттестации по образовательным программам основного общего образования» ГАОУ ДПО МЦКО 2017 год.

"ИК технологии при обучении робототехнике" Городской методический центр Департамента образования города Москвы 2016 год Дизайнер-верстальщик РГСУ 2013 год

«Методика обучения технологии программирования объектов дополненной реальности на Unity 3D». Городской Методический Центр 2023 год

III. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ МОДУЛЯ (организационно-педагогические)

Материально-технические условия реализации программы: Материально-технические условия реализации программы

Обучение по программе реализовано в режиме синхронно-асинхронной работы слушателей в электронной среде. Для проведения синхронных занятий применяется программа видеоконференцсвязи Zoom. Дополнительно для организации работы слушателей могут использоваться Google-сервисы. Асинхронная работа слушателей реализуется на базе электронного курса на платформе электронного обучения LMS Odin.

Требования к оборудованию

Сеть: скорость соединения от 2 Мб/с.

Оборудование для синхронных занятий: персональный компьютер (рекомендуется) / мобильный телефон / планшет; наушники, микрофон и камера (обязательно).

Для работы на платформе электронного обучения LMS Odin рекомендуется использовать персональный компьютер.

Для успешной реализации Программы необходимо, чтобы рабочее место обучающегося и преподавателя включали в себя:

- компьютеры, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др.

периферийное оборудование:

- принтер (черно/белой печати, формата А4);
- устройства для ввода визуальной информации (сканер, цифровой фотоаппарат, web-камера и пр.);
- устройства создания графической информации (графический планшет), использующиеся для создания и редактирования графических объектов, ввода рукописного текста;
- акустические колонки;
- оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер).

программное обеспечение компьютера:

- операционная система семейства MacOS или Windows;
- программа Python,
- файловый менеджер (в составе операционной системы или др.);
- почтовый клиент (в составе операционных систем или др.);
- браузер (в составе операционных систем или др.);
- мультимедиа проигрыватель (в составе операционной системы или др.);
- антивирусная программа;
- программа-архиватор;
- программа-переводчик;
- программа интерактивного общения;
- текстовый редактор;
- растровый графический редактор;
- звуковой редактор;
- редактор Web-страниц.

Учебно-методическое и информационное обеспечение программы:

Методические рекомендации и пособия по изучению курса.

Программа реализуется в формате смешанного обучения. Синхронные занятия включают интерактивные лекции и практические занятия, предусматривающие групповую и индивидуальную работу слушателей. Для организации асинхронной работы слушателей используются записи синхронных занятий, презентации, конспекты лекций, практические задания и тестовые вопросы, размещаемые в электронном курсе на платформе электронного обучения LMS Odin.

Содержание комплекта учебно-методических материалов.

Для изучения программы используется электронный курс в LMS Odin. Электронный курс включает информационно-организационные материалы по программе (учебно-тематический план, календарный график работы по программе, информацию о преподавателях программы, чат для объявлений и вопросов, сведения о результатах обучения), набор видеолекций, презентации к лекциям, набор ссылок на внешние ресурсы, систему заданий с инструкциями, списки основной и дополнительной литературы.

Список литературы

Литература для педагога

1. Марк Лутц - Изучаем Python. 5-е издание
2. Аллен Б. Дауни - Python. Экспресс-курс
3. Марк Саммерфилд - Программирование на Python 3. Подробное руководство

4. Билл Любанович - Изучаем Python. Программирование игр, визуализация данных, веб-приложения
5. Эрик Мэтис - Python для детей. Самоучитель по программированию
6. Харрисон К. Дж. - Python для начинающих. Компьютерные науки на Python
7. Райан Митчелл - Web-разработка на Python. Django
8. Альберт Свейгарт - Автоматизация рутинных задач с помощью Python. Практическое руководство для начинающих
9. Бретт Слаткин - Effective Python. 90 способов написать лучший код
10. Дэвид Бизли - Python. Карманный справочник. Версия 3.7 и выше

Компетенции

Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующие компетенции
строка, не менее 10 символов	строка, значение из "общекультурные", "общепрофессиональные", "профессиональные"	строка не менее 50 символов, перечень знаний	строка не менее 50 символов, перечень умений	строка не менее 50 символов, перечень инструментов
ПК-1 Создание программ на языке Python	общепрофессиональные	<ul style="list-style-type: none"> - Принципы работы информационных технологий - Принципы работы алгоритмов - Существующие структуры данных - Принципы написания программ - Работа с командной строчкой 	<ul style="list-style-type: none"> - Устанавливать Python IDE, Pycharm на ПК - Писать циклы, условия, функции на Python - Подключать сторонние библиотеки 	<ul style="list-style-type: none"> - Средствами разработки на Python Командной строкой (cmd) - Методами машинного обучения - предварительной обработкой данных - подходом к решению задач машинного обучения

			-Подключение библиотек		
--	--	--	------------------------	--	--

Модули

Наименование поля	Допустимые значения	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	целое число	1	2	3	4

Название модуля	1.	Модуль 1: Основы языка Python	Модуль 2: Работа с данными в Python	Модуль 3: Работа с библиотеками Python	Модуль 4: Разработка приложений на Python
-----------------	----	-------------------------------	-------------------------------------	--	---

<p>Описание модуля</p>	<p>2.</p>	<p>Модуль 1. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 1.1 Введение в Python</p> <p>Тема 1.2 Переменные и типы данных</p> <p>Тема 1.3 Управляющие конструкции</p> <p>Тема 1.4 Функции</p> <p>Тема 1.5 Строки и последовательности</p> <p>Тема 1.6 Списки и кортежи</p> <p>Тема 1.7 Словари и множества</p> <p>Тема 1.8 Обработка ошибок и отладка</p> <p>Тема 1.9 Модули и пакеты</p>	<p>Модуль 2. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 2.1 Основы работы с данными</p> <p>Тема 2.2 Чтение и запись данных</p> <p>Тема 2.3 Работа с базами данных</p> <p>Тема 2.4 Обработка и анализ данных</p> <p>Тема 2.5 Визуализация данных</p> <p>Тема 2.6 Работа с API</p> <p>Тема 2.7 Web-скрапинг</p> <p>Тема 2.8 Регулярные выражения</p> <p>Тема 2.9 Работа с XML и JSON</p> <p>Тема 2.10 Работа с CSV и Excel</p>	<p>Модуль 3. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 3.1 NumPy</p> <p>Тема 3.2 Pandas</p> <p>Тема 3.3 Matplotlib</p> <p>Тема 3.4 PyTorch</p> <p>Тема 3.5 Seaborn</p> <p>Тема 3.6 Scikit-learn</p> <p>Тема 3.7 TensorFlow</p> <p>Тема 3.8 BeautifulSoup</p> <p>Тема 3.9 Django</p>	<p>Модуль 4. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 4.1 Основы разработки на Python</p> <p>Тема 4.2 Создание консольных приложений</p> <p>Тема 4.3 Разработка графических интерфейсов</p> <p>Тема 4.4 Разработка веб-приложений</p> <p>Тема 4.5 Разработка игр на Python</p> <p>Тема 4.6 Разработка приложений для мобильных устройств</p> <p>Тема 4.7 Создание API на Python</p>
------------------------	-----------	---	---	---	--

Аттестация по итогам модуля. Количество ак. часов	целое число	1	1	1	1	1
Аттестация по итогам модуля. Формы контроля	строка не менее 4 символов	Зачет	Зачет	Зачет	Зачет	Зачет
Аттестация по итогам модуля. Диагностические инструменты	строка не менее 10 символов	Практическое задание № 1. Основы языка Python	Практическое задание № 2. Работа с данными в Python	Практическое задание № 3. Работа с библиотеками Python	Практическое задание № 4. Разработка приложений на Python (3 ч.)	Практическое задание № 4. Разработка приложений на Python (3 ч.)
Аттестация по итогам модуля. Показатели и критерии оценивания	строка не менее 50 символов	<p>В ходе реализации программы предусмотрены следующие виды оценочных мероприятий:</p> <ul style="list-style-type: none"> - практические задания и упражнения в период синхронной работы; - индивидуальные задания, в том числе со взаимной проверкой (комментированием, рецензированием). 				
Аттестация по итогам модуля. Шкала оценивания, нижнее значение	целое число	2-5	2-5	2-5	2-5	2-5

Аттестация по итогам модуля. Шкала оценивания, верхнее значение	целое число	2-5	2-5	2-5	2-5
Аттестация по итогам модуля. Шкала оценивания, минимальный проходной балл для успешной сдачи	целое число в диапазоне шкалы	3	3	3	3

РАБОЧАЯ ПРОГРАММА

«Python Essentials: концепции и технологии разработки»

I. АННОТАЦИЯ

Программа «Python Essentials: концепции и технологии разработки» Программа направлена на формирование профессиональных практических навыков в области цифровых технологий актуальных в 2023-2024 году. Программа направлена на формирование компетенций в соответствии с трудовыми функциями программиста.

В результате обучения выпускник программы будет способен:

- Понимать основные концепции языка Python, включая переменные, типы данных, управляющие конструкции и функции.
- Научаться создавать и обрабатывать программы на языке Python, используя различные технологии и инструменты, такие как NumPy, Pandas, Matplotlib, Flask и др.
- Научаться работать с различными типами данных, такими как числа, строки, списки, словари и другие.

- Будут уметь использовать управляющие конструкции, такие как условные операторы, циклы и функции, для создания более сложных программ.
- Научатся работать с функциями и модулями, включая создание и использование пользовательских функций и импортирование модулей.
- Будут уметь использовать Python для работы с данными и анализа данных, в том числе использование библиотек Pandas и NumPy.
- Научатся создавать и разрабатывать веб-приложений на языке Python,
- Будут понимать основные принципы программирования, такие как алгоритмы, структуры данных и объектно-ориентированное программирование.
- Будут уметь решать проблемы, используя программирование на языке Python, включая написание и отладку программ.
- Понимать возможности и перспективы использования языка Python в различных областях, таких как наука о данных, машинное обучение, веб-разработка и другие.

Рабочая программа с описанием каждого модуля

Модуль 1.

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 1: Основы языка Python (36 ч) Введение в Python: установка, интерпретатор, IDE Переменные, типы данных,	Тема 1.1 Введение в Python	Введение в Python. Задачи профессии программиста. Сферы применения программирования. Программа как вид информационной модели. Программа как информационный процесс.	теоретические занятия	(1 ч.)
			практические занятия	

операторы, условные выражения и циклы Функции и модули, работа со строками и списками	Тема 1.2 Переменные и типы данных	Переменные и типы данных. Понятие переменной. Имена переменных. Операторы присваивания. Арифметические операции.	самостоятельная работа	(1 ч.)
			практические занятия	
			самостоятельная работа	
	Тема 1.3 Управляющие конструкции	Работа по закреплению материала. Упражнения по использованию циклических конструкций при решении задач на языке Python.	теоретические занятия	(1 ч.)
			практические занятия	
			самостоятельная работа	
	Тема 1.4 Функции	Функции. Работа с многомерными списками данных. Работа с функцией enumerate()	теоретические занятия	(1 ч.)
		Задание к теме 1.4 Работа с функциями	практические занятия	(5 ч)

	Изучить дополнительный материал по теме 1.4 Функции https://metanit.com/python/tutorial/2.8.php	самостоятельная работа	(1 ч.)
Тема 1.5 Строки и последовательности	Строки и последовательности	теоретические занятия	(1 ч.)
	Задание к теме 1.5 Строки и последовательности в Python	практические занятия	(5 ч.)
	Изучить дополнительный материал по теме 1.5. Строки и последовательности https://metanit.com/python/tutorial/5.1.php	самостоятельная работа	(1 ч.)
Тема 1.6 Списки и кортежи	Списки и кортежи/ обавление, удаление, поиск элемента. Работа с кортежем. Распаковка кортежа. Присваивание значения кортежа. Работа с многомерными списками данных.	теоретические занятия	(1 ч.)
	Задание к теме 1.6 Списки и кортежи	практические занятия	(2 ч.)

	Изучить дополнительный материал по теме 1.6. Списки и кортежи https://metanit.com/python/tutorial/3.1.php	самостоятельная работа	(1 ч.)
Тема 1.7 Словари и множества	Словари и множества	теоретические занятия	(1 ч.)
	Задание к теме 1.7 Словари и множества.	практические занятия	(3 ч.)
	Изучить дополнительный материал по теме 1.7 Словари и множества https://metanit.com/python/tutorial/3.3.php	самостоятельная работа	(1 ч.)
Тема 1.8 Обработка ошибок и отладка	Обработка ошибок и отладка	теоретические занятия	(1 ч.)
	Задание к теме 1.8 Обработка ошибок и отладка	практические занятия	(3 ч.)
	Изучить дополнительный материал по теме 1.8. Обработка ошибок и отладка	самостоятельная работа	(1 ч.)

	https://pro.arcgis.com/ru/pro-app/latest/arcpy/get-started/error-handling-with-python.htm	
Тема 1.9 Модули и пакеты	Модули и пакеты . Чтение данных из файла, запись данных в файл.	теоретические занятия (1 ч.)
	Обработка на практике изученной темы. Практическое задание № 1. Основы языка Python.	практические занятия (3 ч.)
	Изучить дополнительный материал по теме 1.9. Модули и пакеты https://metanit.com/python/tutorial/2.10.php	самостоятельная работа (1 ч.)
		Объем в ак.ч.
	ИТОГО:	Объем в % 9
	теоретические занятия	21
	практические занятия	58%
	самостоятельная работа	6 17%

аттестация	1
Всего:	36

Рабочая программа с описанием каждого модуля

Модуль 2

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 2: Объектно-ориентированное программирование на Python Классы и объекты в Python Наследование, полиморфизм и инкапсуляция Работа с файлами	Тема 2.1 Основы работы с данными	Знакомство с понятиями: программа, алгоритм, язык программирования, среда выполнения алгоритма, исполнитель. Сферы применения программирования. Программа как вид информационной модели. Основы работы с данными. Решение практических заданий . Входящие, выходящие, промежуточные данные. Задание к теме 2.1 Основы работы с данными	теоретические занятия практические занятия	(1 ч.) (1 ч.)

и базами данных на Python			самостоятельная работа	
Тема 2.2 Чтение и запись данных	Чтение данных из файла, запись данных в файл. Двоичные файлы. Чтение и запись данных . Устойчивость. Оператор форматирования.		теоретические занятия	(1 ч.)
	Решение практических задач по работе с Чтением и записями данных Задание к теме 2.2		практические занятия	(2 ч.)
Тема 2.3 Работа с базами данных			самостоятельная работа	
	Работа с базами данных		теоретические занятия	(1 ч.)
	Задание к теме 2.3 Решение практических задач по работе с базами данных		практические занятия	(2 ч.)
	Изучить дополнительный материал по теме 2.3 Работа с базами данных https://metanit.com/python/database/		самостоятельная работа	(1 ч.)

	Работа по закреплению материала. Упражнения по работе с базами данных			
Тема 2.4 Обработка и анализ данных	Обработка и анализ данных	теоретические занятия	(1 ч.)	
	Работа по закреплению материала. Решение практических заданий по анализу, исполнению алгоритмов, перевод словесного описания алгоритма в другую форму записи. Упражнения по работе с анализом данных Задание к теме 2.4 Обработка и анализ данных	практические занятия	(2 ч.)	
Тема 2.5 Визуализация данных	Изучить дополнительный материал по теме 2.4 Обработка и анализ данных https://habr.com/ru/articles/353050/	самостоятельная работа	(1 ч.)	
	Визуализация данных	теоретические занятия	(1 ч.)	
	Задание к теме 2.5 Визуализация данных	практические занятия	(2 ч.)	
	Изучить дополнительный материал по теме 2.5 Визуализация данных https://medium.com/nuances-of-programming/5-%D0%BF%D1%80%D0%BE%D1%81%D1	самостоятельная работа	(1 ч.)	

	<p>%82%D1%8B%D1%85-%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BE%D0%B2-%D0%B2%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B8-%D1%86%D0%B8%D0%B8-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-%D0%BD%D0%B0-python-%D1%81-%D0%BA%D0%BE%D0%B4%D0%BE%D0%BC-e0053808c83d</p>		
<p>Тема 2.6 Работа с API</p>	<p>Работа с API . Сбор данных через API. Распространенные форматы текстовых файлов</p>	<p>теоретические занятия</p>	<p>(1 ч.)</p>
	<p>Задание к теме 2.6 Работа с API. Решение практических задач по сбору данных через API. Разбор формата JSON. Решение задач на API VK. Отработка навыков решения задач уровня middle – веб-краулинг с помощью API</p>	<p>практические занятия</p>	<p>(3 ч.)</p>
		<p>самостоятельная работа</p>	
	<p>Web-скрапинг. Скрапинг данных. Скрапинг табличных данных.</p>	<p>теоретические занятия</p>	<p>(1 ч.)</p>

Тема 2.7 Web-скрапинг	Задание к теме 2.7 Web-скрапинг . Отработка навыков решения задач	практические занятия	(2 ч.)
		самостоятельная работа	
Тема 2.8 Регулярные выражения	Регулярные выражения. Решение практических задач по работе с условными выражениями.	теоретические занятия	(1 ч.)
	Задание к теме 2.8 Регулярные выражения . Работа по закреплению материала. Упражнение по составлению условий .	практические занятия	(2 ч.)
Тема 2.9 Работа с XML и JSON	Изучить дополнительный материал по теме 2.8 Регулярные выражения https://habr.com/ru/articles/349860/	самостоятельная работа	(1 ч.)
	Работа с XML и JSON	теоретические занятия	(1 ч.)
	Задание к теме 2.9 Работа с XML и JSON	практические занятия	(2 ч.)
	Изучить дополнительный материал по теме 2.9 Работа с XML и JSON	самостоятельная работа	(1 ч.)

		https://pythonim.ru/osnovy/preobrazovanie-xml-v-json-i-dict-v-python			
Тема 2.10 Работа с CSV и Excel		Распространенные форматы текстовых файлов: CSV, JSON.	теоретические занятия	(3 ч.)	
		Отработать изученный материал с помощью заданий . Практическое задание № 2. Roblox Studio	практические занятия		
		Изучить дополнительный материал по теме 2.10 Работа с CSV и Excel https://pythonim.ru/osnovy/rabota-s-csv-faylami-python	самостоятельная работа	(1 ч.)	
				Объем в ак.ч.	Объем в %
ИТОГО:			теоретические занятия	9	
			практические занятия	21	58%
			самостоятельная работа	6	17%

аттестация	1
Всего:	36

Рабочая программа с описанием каждого модуля

Модуль 3

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 3: Работа с библиотеками Python	Тема 3.1 NumPy	Библиотека Numpy. Одномерные и многомерные массивы данных. Операции над массивами данных. Методы массивов. Индексация. (NumPy.)	теоретические занятия	(1 ч.)
Введение в библиотеки Python: NumPy, Pandas,		Решение практических задач на сложение, вычитание, умножение и сортировку массивов. Задачи на основные операции с массивами: объявление, заполнение, вывод на экран; обработку элементов массивов. Задание к теме 3.1 NumPy	практические занятия	(2 ч.)

Matplotlib, Seaborn Работа с данными, их обработка и визуализация	<p>Работа по закреплению материала. Упражнения по работе с массивами данных с помощью библиотеки NumPy. Изучить дополнительный материал по теме 3.1 NumPy https://habr.com/ru/articles/352678//</p>	самостоятельная работа	(1 ч.)
	<p>Работа с Pandas</p>	теоретические занятия	(1 ч.)
Работа с API и web-сервисами на Python	<p>Задание к теме 3.2 Работа с Pandas</p>	практические занятия	(1 ч.)
	<p>Изучить дополнительный материал по теме 3.2Pandas https://khashtamov.com/ru/pandas-introduction/</p>	самостоятельная работа	(1 ч.)
	<p>Тема 3.3 Matplotlib</p>	теоретические занятия	(1 ч.)
	<p>Отработать полученные знания на практике. Задание к теме 3.3 Работа с Matplotlib</p>	практические занятия	(2 ч.)

	Изучить дополнительный материал по теме 3.3 Matplotlib https://pythonworld.ru/novosti-mir-python/scientific-graphics-in-python.html	самостоятельная работа	(1 ч.)
Тема 3.4 PyTorch	Обучение модели анализа данных с помощью PyTorch. Параметры модели. Принципы работы. Изучить на практике полученный материал, отработать практические навыки с помощью задания. Практическое задание № 3. Создание анимации в Pygame	теоретические занятия практические занятия	(1 ч.) (3 ч.)
	Изучить дополнительный материал по теме 3.4 PyTorch https://habr.com/ru/articles/334380/	самостоятельная работа	(1 ч.)
Тема 3.5 Seaborn	Работа с Seaborn . Библиотека для создания статистических графиков на Python	теоретические занятия	(1 ч.)

	<p>Закрепление изученного материала на практике Задание к теме 3.5 Работа с Seaborn</p>	практические занятия	(2 ч.)
	самостоятельная работа		
Тема 3.6 Scikit-learn	<p>Библиотека Scikit-learn: как создать свой первый ML-проект</p>	теоретические занятия	(1 ч.)
	<p>Отработка практической части, по изученному материалу. Задание к теме 3.6 Работа с Scikit-learn</p>	практические занятия	(3 ч.)
	самостоятельная работа		
Тема 3.7 TensorFlow	<p>Изучение теоретического материала- TensorFlow .</p>	теоретические занятия	(1 ч.)
	<p>На практике показываем, как работает TensorFlow Задание к теме 3.7 Работа с TensorFlow</p>	практические занятия	(3 ч.)

	самостоятельная работа		
Тема 3.8 BeautifulSoup	теоретические занятия	<p>Введение в Requests. Углубленный анализ HTML кода. HTML. Атрибуты http-запросы, html-страницы. Метод requests.get(). Статус коды. Получаем содержимое response объекта.</p>	(1 ч.)
	практические занятия	<p>Решение практических задач по работе с библиотеками requests и BeautifulSoup. Работа с протоколом HTTP. Задание к Теме 3.8 Работа с BeautifulSoup</p>	(2 ч.)
	самостоятельная работа	<p>Изучить дополнительный материал по теме 3.8 BeautifulSoup https://dvmn.org/encyclopedia/modules/bs4-tutorial/</p>	(1 ч.)
Тема 3.9 Django	теоретические занятия	<p>Изучение теоретического материала Django фреймворк для веб-разработки, написанный на Python. Позволяет</p>	(2 ч.)

	создавать сайты и веб-приложения разной сложности.		
	Отработка практических навыков изученного материала . Задание к Теме 3.9 Тема 3.9 Работа с Django	практические занятия	(2 ч.)
	Изучить дополнительный материал по теме 3.9 Разработка концепции игры https://metanit.com/python/django/	самостоятельная работа	(1 ч.)
			Объем в ак.ч.
			Объем в %
	ИТОГО:	теоретические занятия	10
		практические занятия	20 57%
		самостоятельная работа	6 17%
		аттестация	1

Всего:	36
---------------	-----------

Рабочая программа с описанием каждого модуля

Модуль 4

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
Модуль 4: Разработка приложений на Python	Тема 4.1 Основы разработки на Python	Основы программирования на языке Python. Задание к теме 4.1 Основы разработки на Python	теоретические занятия	(1 ч.)
			практические занятия	(3 ч.)
Разработка GUI приложений на Python с помощью PyQt5		Изучить дополнительный материал по теме 4.1 Основы разработки на Python https://umschool.net/library/informatika/osnovy-programirovaniya-na-yazyuke-python-chast-1/	самостоятельная работа	(1 ч.)
Введение в веб-разработку на	Тема 4.2 Создание	Разбор компьютерная программы (приложения или утилиты),	теоретические занятия	(2 ч.)

<p>Python: Flask и Django</p> <p>Пакетирование и распространение приложений на Python</p>	<p>консольных приложений</p>	<p>предназначенной для использования через текстовый пользовательский интерфейс, такой как текстовый терминал, интерфейс командной строки некоторых операционных систем . Создание консольных приложений</p>	
		<p>Отработка изученного материала на практике. Задание к теме 4.2 Создание консольных приложений</p>	<p>практические занятия</p> <p>(2 ч.)</p>
			<p>самостоятельная работа</p>
<p>Тема 4.3 Разработка графических интерфейсов</p>		<p>Решение практических задач по работе с графической областью для рисования, рисование разного рода линий. Рисование геометрических фигур и картинок из геометрических фигур. Цвет линии, цвет заливки, раскрашивание нарисованных картинок.</p>	<p>теоретические занятия</p> <p>(1 ч.)</p>
		<p>Отработка полученных знаний на практике Задание к теме 4.3 Разработка графических интерфейсов</p>	<p>практические занятия</p> <p>(3 ч.)</p>
		<p>Изучить дополнительный материал по теме 4.3 Разработка графических</p>	<p>самостоятельная работа</p> <p>(1 ч.)</p>

	интерфейсов https://habr.com/ru/companies/e-dison/articles/480884/		
Тема 4.4 Разработка веб-приложений	<p>Разработка веб-приложений . Создание веб-приложение на Python с использованием фреймворка Flask</p> <p>Задание к теме 4.4 Разработка веб-приложений Проработка изученного. Решение практических задач по созданию веб-приложений</p> <p>Изучить дополнительный материал по теме 4.4 Разработка веб-приложений https://habr.com/ru/companies/ruvds/articles/507778/</p>	<p>теоретические занятия (1 ч.)</p> <p>практические занятия (2 ч.)</p> <p>самостоятельная работа (1 ч.)</p>	
Тема 4.5 Разработка игр на Python	<p>Разработка игр на Python Создание игры "Крестики-нолики" на Python, используя консольный интерфейс</p> <p>Отработка теории на практике . Задание к теме 4.5 Разработка игр на Python</p>	<p>теоретические занятия (1 ч.)</p> <p>практические занятия (3 ч.)</p>	

	Изучить дополнительный материал по теме 4.5 Разработка игр на Python https://habr.com/ru/articles/645041/	самостоятельная работа	(1 ч.)
Тема 4.6 Разработка приложений для мобильных устройств	Учимся разрабатывать на языке Python с использованием фреймворка Pygame. Разработка приложений для мобильных устройств	теоретические занятия	(2 ч.)
	Отработка полученных знаний на практике Задание к теме 4.6 Разработка приложений для мобильных устройств	практические занятия	(3 ч.)
	Изучить дополнительный материал по теме 4.6 Разработка приложений для мобильных устройств https://proglib.io/p/mobile-python	самостоятельная работа	(1 ч.)
Тема 4.7 Создание API на Python	Создание API на Python	теоретические занятия	(1 ч.)
	Практическое задание № 4. Разработка приложений на Python	практические занятия	(2 ч.) (1 асс. ч.)

		Изучить дополнительный материал по теме 4.7 Создание API на Python https://habr.com/ru/companies/skillbox/articles/464705/	самостоятельная работа	(1 ч.)	
Итоговая аттестация			Итоговая проектная работа	(2 ч.)	
				Объем в ак.ч.	Объем в %
ИТОГО:			теоретические занятия	9	
			практические занятия	19	53%
			самостоятельная работа	6	17%
			промежуточная аттестация	1	
			итоговая аттестация	2	
Всего:				36	

Задание к теме 1.4

Работа с функциями (5 ч)

1. Откройте пустой проект в IDLE
2. Создает функцию, которая принимает на вход два аргумента - числа a и b, и возвращает их сумму
3. Создает функцию, которая принимает на вход два аргумента - числа a и b, и возвращает их произведение.
4. Создает функцию, которая принимает на вход список чисел и возвращает среднее значение всех элементов списка.
5. Создает функцию, которая принимает на вход список чисел и возвращает максимальное значение в списке.
6. Создает функцию, которая принимает на вход строку и возвращает ее в обратном порядке.

Далее в основной программе:

1. Запрашивает у пользователя два числа и выводит на экран их сумму и произведение, используя соответствующие функции.
2. Запрашивает у пользователя список чисел, вызывает функции для нахождения среднего значения и максимального значения в списке, и выводит эти значения на экран.
3. Запрашивает у пользователя строку, вызывает функцию для ее обращения и выводит результат на экран.

Задание к теме 1.5

Строки и последовательности в Python (5 ч.)

1. Откройте пустой проект в IDLE
2. Напишем программу, которая принимает две строки от пользователя и выводит их конкатенацию

Конкатенация строк - это одна из базовых операций при работе со строками в Python. Для выполнения этой операции используется оператор "+". В этой программе мы сначала запрашиваем у пользователя две строки при помощи функции input(). Затем мы объединяем эти две строки в одну строку, используя оператор "+", и сохраняем результат в переменную concat_str. Наконец, мы выводим результат на экран при помощи функции print().

Задание к теме 1.6

Списки и кортежи (4 ч.)

1. Откройте пустой проект в IDLE
2. Создайте список, состоящий из элементов разных типов данных (целое число, вещественное число, строка, логическое значение).

3. Выведите на экран первый и последний элементы списка.
4. Измените значение третьего элемента списка на новую строку.
5. Добавьте в конец списка новый элемент - целое число.
6. Удалите из списка второй элемент.
7. Создайте кортеж из элементов списка.
8. Выведите на экран длину кортежа.
9. Выведите на экран второй элемент кортежа.
10. Попробуйте изменить значение второго элемента кортежа. Объясните результат.
11. Создайте новый список из элементов кортежа.
12. Выведите на экран типы данных всех элементов нового списка.

Задание к теме 1.7

Словари и множества (3 ч.)

1. Откройте пустой проект в IDLE
2. Создайте словарь, содержащий информацию о трех ваших любимых фильмах. Ключом словаря должно быть название фильма, а значением - список, содержащий год выпуска фильма, имя режиссера и жанр. Выведите на экран всю информацию о фильмах в формате "Название фильма (год выпуска), режиссер - Имя режиссера, жанр - Жанр".

Задание к теме 1.8

Обработка ошибок и отладка. (3 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу на Python, которая запрашивает у пользователя два числа и выводит результат их деления. Однако, необходимо предусмотреть возможность ошибки при делении на ноль. Если пользователь попытается разделить на ноль, программа должна вывести сообщение об ошибке и запросить ввод двух чисел снова.

Задание к теме 2.1

Основы работы с данными (1 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу на Python, которая запрашивает у пользователя его имя и возраст, а затем выводит сообщение с приветствием и возрастом пользователя, округленным до целого числа.

Задание к теме 2.2

Чтение и запись данных (2 ч.)

1. Откройте пустой проект в IDLE
2. Написать программу на Python, которая просит пользователя ввести своё имя и возраст, а затем записывает эти данные в файл "info.txt".
После этого программа должна прочитать данные из файла и вывести их на экран.

Задание к теме 2.3

Работа с базами данных (2 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу на Python, которая подключается к базе данных SQLite, создает новую таблицу с именем "users" и добавляет в нее две записи. Первая запись должна содержать имя "John" и возраст 25 лет, вторая запись должна содержать имя "Jane" и возраст 30 лет.
После добавления записей, программа должна выбрать все записи из таблицы "users" и вывести их на экран.

Задание к теме 2.4

Обработка и анализ данных (2 ч.)

1. Откройте пустой проект в IDLE
2. Вам необходимо написать программу на Python, которая принимает на вход текстовый файл с данными о продажах продуктов в магазине, обрабатывает этот файл и выводит на экран следующую информацию:
 - Общее количество проданных товаров
 - Общая выручка магазина
 - Средняя стоимость продажи одного товара

- Топ-3 самых продаваемых товаров

Формат файла:

Каждая строка файла содержит информацию о продаже одного товара. В строке содержится название товара, количество проданных единиц товара и цена за единицу товара, разделенные пробелами. Пример строки: "Хлеб 5 25.5"

Задание к теме 2.5

Визуализация данных (2 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу, которая загружает данные о продажах продуктов в формате CSV и визуализирует их в виде графика. Данные должны содержать следующие поля: название продукта, количество проданного товара, дата продажи.
 - Необходимо загрузить библиотеку Pandas и matplotlib.
 - Загрузите данные из CSV файла в Pandas DataFrame.
 - Используйте методы Pandas для группировки продаж по продукту и дате.
 - Создайте график, отображающий количество продаж каждого продукта на оси Y и даты на оси X.
 - Добавьте заголовки и метки к осям.
 - Сохраните график в формате PNG.

Задание к теме 2.6

Работа с API (3 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу, которая получает информацию о погоде с помощью API и выводит ее на экран. Для этого вам понадобится использовать библиотеку requests.
3. Получите API-ключ для сервиса погоды, например, OpenWeatherMap.

4. Загрузите библиотеку `requests`.
5. Используйте функцию `requests.get()` для получения данных с API. При этом необходимо передать URL-адрес запроса и API-ключ.
6. Извлеките нужную информацию из ответа API и выведите ее на экран.

Задание к теме 2.7

Web-скрапинг (2 ч.)

1. Откройте пустой проект в IDLE

2. Напишите программу, которая собирает информацию о курсах валют с сайта Центробанка России и выводит ее в удобном формате. Необходимо собрать данные о текущем курсе доллара, евро и фунта стерлингов. Для этого можно использовать библиотеку `BeautifulSoup` для парсинга HTML страницы и `requests` для отправки запроса на сервер.

Задание к теме 2.8

Регулярные выражения (2 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу, которая запрашивает у пользователя строку и проверяет, содержит ли эта строка только цифры.

Задание к теме 2.9

Работа с XML и JSON (2 ч.)

1. Откройте пустой проект в IDLE
2. Напишем программу на Python, которая считывает данные из файла в формате XML или JSON и выводит их на экран. Для примера будем использовать файл в формате JSON.

Задание к теме 3.1

NumPy (2 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу, которая создает массив NumPy из 10 случайных целых чисел от 1 до 100 и выводит его на экран. Затем найдите среднее значение массива и выведите его на экран.

Задание к теме 3.2

Работа с Pandas (1 ч.)

1. Откройте пустой проект в IDLE
2. Загрузите данные из CSV-файла в DataFrame с помощью функции `read_csv()`. Файл должен содержать информацию о погоде за неделю.
3. Отобразите первые 5 строк DataFrame с помощью функции `head()`.
4. Используя метод `describe()`, выведите статистическую информацию о DataFrame.
5. Отобразите только те строки DataFrame, где температура была выше средней за неделю.
6. Добавьте новый столбец Тип погоды, который будет содержать значения "Солнечно", "Облачно" или "Дождливо" в зависимости от значений столбца Описание погоды.
7. Сохраните DataFrame в CSV-файл с помощью функции `to_csv()`.

Задание к теме 3.3

Работа с Matplotlib (2 ч.)

1. Откройте пустой проект в IDLE
2. Постройте график линейной функции используя Matplotlib на Python

Задание к теме 3.5

Работа с Seaborn (2 ч.)

1. Откройте пустой проект в IDLE
2. Импортируйте библиотеки Seaborn и Pandas
3. Загрузите данные о расходах на рекламу из файла "advertising.csv" в DataFrame с помощью Pandas
4. Используйте метод `pairplot()` из Seaborn, чтобы создать матрицу графиков рассеяния для всех пар переменных
5. Добавьте аргумент `hue`, чтобы отобразить точки разных классов в разных цветах. В нашем случае мы будем использовать столбец "Clicked on Ad"
6. Используйте метод `heatmap()` из Seaborn, чтобы построить тепловую карту корреляции между всеми парами переменных
7. Добавьте аргумент `annot=True`, чтобы отобразить значения корреляции в ячейках тепловой карты.

Задание к теме 3.6

Работа с Scikit-learn (3 ч.)

1. Откройте пустой проект в IDLE
2. Начните с загрузки набора данных изображений, например, набора данных MNIST, который включен в библиотеку Scikit-learn.
3. Разделите набор данных на тренировочную и тестовую выборки, используя функцию `train_test_split()`.
4. Создайте экземпляр модели машинного обучения, например, логистической регрессии, используя класс `LogisticRegression` из библиотеки Scikit-learn.
5. Обучите модель на тренировочных данных, используя метод `fit()`.
6. Оцените качество модели на тестовых данных, используя метод `score()`.
7. Попробуйте улучшить модель, изменяя гиперпараметры модели, например, C-параметр в логистической регрессии.
8. После улучшения модели повторите шаги 4-5, чтобы оценить ее новое качество.
9. Наконец, используйте модель для предсказания классов новых изображений, используя метод `predict()`.

Задание к теме 3.7

Работа с TensorFlow (3 ч.)

1. Откройте пустой проект в IDLE
2. Написать программу, используя TensorFlow, которая будет классифицировать изображения на два класса: собаки и кошки. Для этого необходимо:
 3. Скачать набор данных изображений собак и кошек.
 4. Разделить данные на обучающую и тестовую выборки.
 5. Загрузить данные в TensorFlow.
 6. Создать нейронную сеть, которая будет обучаться на обучающей выборке.
 7. Оценить точность модели на тестовой выборке.
 8. Протестировать модель на собственных изображениях собак и кошек.
9. В качестве примера можно использовать набор данных с изображениями собак и кошек, доступный на сайте Kaggle:
<https://www.kaggle.com/c/dogs-vs-cats/data>.

Задание к теме 3.8

Работа с BeautifulSoup (2 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу на Python, использующую библиотеку BeautifulSoup для работы с HTML-кодом веб-страницы.

Задачи программы:

- Загрузить веб-страницу с помощью библиотеки requests.
- Используя BeautifulSoup, получить текст всех заголовков h1 и вывести их на экран.
- Получить все ссылки на странице и вывести их на экран.
- Получить текст первого абзаца (тэг p) и вывести его на экран.
- Найти все картинки на странице и сохранить их на жесткий диск.

Примерный план решения:

- Импортировать библиотеки requests и BeautifulSoup.
- Задать URL-адрес страницы.
- С помощью requests получить содержимое страницы.

- Создать объект BeautifulSoup на основе полученного HTML-кода.
- Получить все заголовки h1 с помощью метода find_all и вывести их на экран.
- Получить все ссылки с помощью метода find_all и вывести их на экран.
- Получить первый абзац с помощью метода find и вывести его на экран.
- Получить все картинки с помощью метода find_all и сохранить их на жесткий диск.

Задание к теме 3.9

Работа с Django (2 ч.)

1. Откройте пустой проект в IDLE
2. Создайте простое веб-приложение, которое будет отображать список пользователей. Каждый пользователь должен иметь имя, фамилию и адрес электронной почты. Приложение должно иметь две страницы:
 - Главная страница должна отображать список всех пользователей в базе данных. Каждый пользователь должен быть представлен в виде ссылки, которая ведет на страницу с деталями пользователя.
 - Страница деталей пользователя должна отображать имя, фамилию и адрес электронной почты пользователя.

Вы можете использовать следующие инструменты и технологии:

Django 3.x и **Python 3.x**

SQLite база данных

HTML, CSS и **JavaScript** для шаблонов и отображения данных.

Задание к теме 4.1

Основы разработки на Python (3 ч.)

1. Откройте пустой проект в IDLE
2. Напишите программу на Python, которая запрашивает у пользователя его имя и возраст, а затем выводит сообщение с приветствием и информацией о том, сколько лет пользователю исполнилось или сколько ему исполнилось в текущем году. Для решения задачи понадобятся функции input(), int(), str(), а также операторы для работы со строками и арифметическими выражениями.

Задание к теме 4.2

Создание консольных приложений (2 ч.)

1. Откройте пустой проект в IDLE
2. Создайте консольное приложение на Python, которое запрашивает у пользователя его имя и возраст, а затем выводит сообщение вида "Привет, {имя}! Тебе уже {возраст} лет."

Задание к теме 4.3

Разработка графических интерфейсов (3 ч.)

1. Откройте пустой проект в IDLE
2. Создайте графический интерфейс на Python с использованием библиотеки Tkinter, который позволяет пользователю ввести своё имя и возраст, а затем выводит сообщение вида "Привет, {имя}! Тебе уже {возраст} лет."

Задание к теме 4.4

Разработка веб-приложений (2 ч.)

1. Откройте пустой проект в IDLE
2. Создайте веб-приложение на Python с использованием фреймворка Flask, которое позволяет пользователю ввести своё имя и возраст, а затем выводит сообщение вида "Привет, {имя}! Тебе уже {возраст} лет."

Задание к теме 4.5

Разработка игр на Python (3 ч.)

1. Откройте пустой проект в IDLE
2. Задание: Создайте игру "Крестики-нолики" на Python, используя консольный интерфейс.

Шаги:

- Создайте пустое поле для игры, состоящее из 9 клеток.
- Напишите функцию, которая будет отображать текущее состояние поля.
- Напишите функцию, которая будет проверять, есть ли выигрышная комбинация на поле.
- Напишите функцию, которая будет проверять, свободна ли выбранная клетка.

- Напишите функцию, которая будет принимать ходы от игроков и записывать их на поле.
- Создайте основной цикл игры, который будет сменять ходы между игроками, проверять выигрышную комбинацию и выводить результаты игры.

Задание к теме 4.6

Разработка приложений для мобильных устройств (3 ч.)

1. Откройте пустой проект в IDLE

Задание: Разработать приложение для мобильных устройств на языке Python с использованием фреймворка Pygame.

Требования:

- Приложение должно быть игрой для мобильного устройства.
- В игре должна быть реализована как минимум одна игровая механика (например, управление персонажем, стрельба, сбор предметов и т.д.).
- В игре должен быть реализован пользовательский интерфейс, включающий в себя как минимум один игровой экран и элементы управления (например, кнопки, ползунки и т.д.).
- Игра должна быть отлажена и проверена на мобильном устройстве.
- Игра должна быть задокументирована с использованием стандартной документации Python.

Рекомендуемые шаги для выполнения задания:

- Изучить фреймворк Pygame и его основные возможности.
- Спроектировать игру, определить игровые механики и интерфейс.
- Реализовать функциональность игры с помощью Pygame.
- Протестировать игру на мобильном устройстве и отладить ошибки.

Календарно-тематическое планирование

Календарный график

Первый модуль с 01.10.2023 - 30.11.2023

Второй модуль: с 01.12.2023 - 31.01.2024

Третий модуль: с 01.02.2024 - 31.03.2024

Четвертый модуль: с 01.04.2024 - 31.05.2024

Занятия проводятся 2-3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

№	Тема и № модуля	Тема занятия	Кол-во занятий*	Кол-во часов	Дата
1	Модуль 1: Основы языка Python 01.10.2023 - 30.11.2023	Тема 1.1 Введение в Python	1	1	02.10.2023
2		Тема 1.2 Переменные и типы данных	1	1	05.10.2023
3		Тема 1.3 Управляющие конструкции	1	1	05.10.2023
4		Тема 1.4 Функции	3	6	09.10.2023-19.10.2023

5		Тема 1.5 Строки и последовательности	3	6	23.10.2023-30.10.2023
6		Тема 1.6 Списки и кортежи	2	3	06.11.2023-09.11.2023
7		Тема 1.7 Словари и множества	2	4	13.11.2023-17.11.2023
8		Тема 1.8 Обработка ошибок и отладка	2	4	20.11.2023-24.11.2023
9		Тема 1.9 Модули и пакеты	2	4	27.11.2023-30.11.2023
1	Модуль 2: Работа с данными в Python 01.12 2023 -31.01.2024	Тема 2.1 Основы работы с данными	2	2	01.12.2023
2		Тема 2.2 Чтение и запись данных	2	3	04.12.2023-06.12.2023
3		Тема 2.3 Работа с базами данных	2	3	08.12.2023-11.12.2023
4		Тема 2.4 Обработка и анализ данных	2	3	14.12-2023-18.12.2023

5				2	3	21.12.2023-25.12.2023
		Тема 2.5 Визуализация данных				
6			2	4	27.12.2023-28.12.2023	
		Тема 2.6 Работа с API				
7			2	3	11.01.2024-15.01.12.2024	
		Тема 2.7 Web-скрапинг				
8			2	3	17.01.2024-19.01.2024	
		Тема 2.8 Регулярные выражения				
9			2	3	22.01.2024-24.01.2024	
		Тема 2.9 Работа с XML и JSON				
10			2	3	29.01.2024-31.01.2024	
		Тема 2.10 Работа с CSV и Excel				
1	Модуль 3: Работа с библиотеками Python 01.02.2024 - 31.03.2024		2	3	01.02.2024-03.02.2024	
		Тема 3.1 NumPy				
2			2	3	05.02.2024-07.02.2024	
		Тема 3.2 Pandas				
3			2	3	15.02.2024-17.02.2024	
		Тема 3.3 Matplotlib				

4		Тема 3.4 PyTorch	2	3	19.02.2024-21.02.2024
5		Тема 3.5 Seaborn	2	3	23.02.2024-26.02.2024
6		Тема 3.6 Scikit-learn	2	4	28.02.2024-01.03.2024
7		Тема 3.7 TensorFlow	2	4	04.03.2024-06.03.2024
8		Тема 3.8 BeautifulSoup	2	3	11.03.2024-14.2024
9		Тема 3.9 Django	2	4	18.03.2024-21.03.2024
		Модуль 4: Разработка приложений на Python	2	4	03.04.2024-06.04.2024
1		Тема 4.1 Основы разработки на Python	2	4	08.04.2024-10.04.2024
		Тема 4.2 Создание консольных приложений	2	4	12.04.2024-17.04.2024
2		Тема 4.3 Разработка графических интерфейсов	2	4	

3		Тема 4.4 Разработка веб-приложений	2	3	19.04.2024-22.04.2024
4		Тема 4.5 Разработка игр на Python	2	4	26.04.2024-29.04.2024
5		Тема 4.6 Разработка приложений для мобильных устройств	4	5	06.05.2024-13.05.2024
6		Тема 4.7 Создание API на Python	2	4	15.05.2024-20.05.2023
7		Итоговая аттестация	2	2	22.05.2024

*количество занятий не включает часы, отведенные на самостоятельное изучение, и часы, отведенные на прохождение аттестации

Учебно-методические материалы

Наименование поля	Допустимые значения полей	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	строка не менее 10 СИМВОЛОВ	1	2	3	4
Методы, формы и технологии	строка не менее 10 СИМВОЛОВ	<p>Программа реализуется в формате смешанного обучения. Синхронные занятия включают интерактивные лекции и практические занятия, предусматривающие групповую и индивидуальную работу слушателей. Для организации асинхронной работы слушателей используются записи синхронных занятий, презентации, конспекты лекций, практические задания и тестовые вопросы, размещаемые в электронном курсе на платформе электронного обучения LMS Odin.</p>			
Методические разработки	строка не менее 10 СИМВОЛОВ				
Материалы модуля	строка не менее 10 СИМВОЛОВ	<p>Для изучения программы используется электронный курс в LMS Odin. Электронный курс включает информационно-организационные материалы по программе (учебно-тематический план, календарный график работы по программе, информацию о преподавателях программы, час для объявлений и вопросов, сведения о результатах обучения), набор видеолекций, презентации к лекциям, набор ссылок на внешние ресурсы, систему заданий с инструкциями, списки основной и дополнительной литературы.</p>			

Учебная литература	строка не менее 10 символов	<ol style="list-style-type: none"> 1. Марк Лутц - Изучаем Python. 5-е издание 2. Аллен Б. Дауни - Python. Экспресс-курс 3. Марк Саммерфилд - Программирование на Python 3. Подробное руководство 4. Билл Любанович - Изучаем Python. Программирование игр, визуализация данных, веб-приложения 5. Эрик Мэлиз - Python для детей. Самоучитель по программированию 6. Харрисон К. Дж. - Python для начинающих. Компьютерные науки на Python 7. Райан Митчелл - Web-разработка на Python. Django 8. Альберт Свейгарт - Автоматизация рутинных задач с помощью Python. Практическое руководство для начинающих 9. Бретт Слаткин - Effective Python. 90 способов написать лучший код 10. Дэвид Бизли - Python. Карманный справочник. Версия 3.7 и выше
--------------------	-----------------------------	---

Материально-технические условия реализации программы

Наименование поля	Допустимые значения полей	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	строка не менее 10 символов	1	2	3	4

<p>Наименование требуемого оборудования</p>	<p>строка не менее 2 символов</p>	<p>- персональный компьютер (рекомендуется) / мобильный телефон / планшет; наушники, микрофон и камера (обязательно); -сеть: скорость соединения от 2 Мб/с.</p>			
<p>Наименование требуемого программного обеспечения</p>	<p>строка не менее 2 символов</p>	<p>операционная система семейства MacOS или Windows;</p> <ul style="list-style-type: none"> ● программа Python; ● файловый менеджер (в составе операционной системы или др.); ● почтовый клиент (в составе операционных систем или др.); ● браузер (в составе операционных систем или др.); ● мультимедиа проигрыватель (в составе операционной системы или др.); ● антивирусная программа; ● программа-архиватор; ● программа-переводчик; ● программа интерактивного общения; ● текстовый редактор; ● растровый графический редактор; ● звуковой редактор; ● редактор Web-страниц. 			
<p>Электронные информационные ресурсы</p>	<p>строка не менее 10 символов</p>				
<p>Электронные образовательные ресурсы</p>	<p>строка не менее 10 символов</p>				