

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 516
НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА

ПРИНЯТА

Педагогический совет
Образовательного учреждения
Протокол № 7 от 24.05.2024 г.

УТВЕРЖДЕНА

Директор

Л.В. Смирнова

Приказ № 84-у от 27.05.2024 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА «ОСНОВЫ
ПРОГРАММИРОВАНИЯ PYTHON»

Срок реализации: 2 года
Возраст обучающихся: 14-17 лет

Разработчик: Бутко Маргарита Михайловна
Педагог дополнительного образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования. Python» (далее - программа) имеет **техническую направленность**.

Адресат программы. Программа рассчитана для детей от 14 до 17 лет.

Актуальность программы. Воспитанники, прошедшие обучение по данной программе, получают знания, умения и навыки, необходимые для сознательного выбора в будущем профессии, связанной с программированием. Программа позволяет школьникам познакомиться и понять тонкости профессии программист и профессий связанных с разработкой IT-приложений, оценить себя в этих профессиях, выработать профессионально важные качества.

Уровень освоения: общекультурный.

Объем ДОП: 144 часа.

Срок освоения: 2 года.

Цель программы: развитие логического мышления, личности обучающегося, способного к творческому самовыражению через овладение основами программирования на одном из самых востребованных языков программирования Python.

Задачи:

Обучающие:

- познакомить с возможностями и особенностями современного языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных;
- научить применять функции при написании программ на языке программирования Python;
- сформировать систему знаний, умений и навыков, необходимых в работе программиста;
- обучить коммуникативным навыкам при совместной работе над проектом.

Развивающие:

- развить познавательные потребности и способности школьников;
- развить творческие способности, логическое мышление;
- развить внимательность и наблюдательность, привить навыки аккуратности и точности в работе;
- замотивировать к постоянному самообразованию.

Воспитательные:

- воспитать навыки взаимодействия при командной работе над проектом;
- воспитать чувство ответственности за результаты деятельности;
- сформировать осознанное отношение к выбору будущей профессии.

Планируемые результаты:

Предметные:

- будут ознакомлены с возможностями и особенностями современного языка программирования Python;
- будут изучены основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных;
- будут обучены применять функции при написании программ на языке программирования Python;
- будет сформирована система знаний, умений и навыков, необходимых в работе программиста;

Личностные:

- будут воспитаны навыки взаимодействия при командной работе над проектом;

- будет воспитано чувство ответственности за результаты деятельности;
- будет сформировано осознанное отношение к выбору будущей профессии.

Метапредметные:

- будут развиты познавательные потребности и способности школьников;
- будут развиты творческие способности, логическое мышление;
- будут развиты внимательность и наблюдательность, привиты навыки аккуратности и точности в работе;
- будут замотивированы к постоянному самообразованию.

Организационно-педагогические условия реализации

Язык реализации – русский.

Форма обучения - очная.

Условия набора в коллектив и формирования групп. Принимаются все желающие. Занятия проводятся в группах, подгруппах и индивидуально, сочетая принцип группового обучения с индивидуальным подходом.

Количество учащихся в группе: 15 человек.

Формы организации занятия. В данной программе используется индивидуальная, групповая и фронтальная формы работы. Предусматривается только аудиторные занятия.

Формы проведения занятий. Лекция, беседа, демонстрация, практические занятия, творческая работа.

Формы организации деятельности учащихся на занятии. Предусматриваются фронтальная и индивидуальная формы.

Материально-техническое обеспечение программы:

- рабочие места, оснащенные персональными компьютерами или ноутбуками с установленным программным обеспечением, находящемся в свободном доступе, - средой программирования Python – 15 шт;
- рабочее место преподавателя, оснащенное персональным компьютером или ноутбуком с установленным программным обеспечением – 1 шт;
- меловая доска – 1 шт;
- проектор – 1 шт;
- комплект учебно-методической документации: рабочая программа кружка, раздаточный материал, задания, цифровые компоненты учебно-методических комплексов (презентации).

Учебно-тематический план 1 года обучения

№ п/п	Раздел (или тема) учебно-тематического плана	Количество часов			Формы контроля
		Теория	Практика	Всего	
1.	Вводное занятие. Техника безопасности.	1	1	2	Беседа.
Раздел 1. Введение в курс (2 ч.)					
2.	История языков программирования. Язык Python	2	2	4	Тестирование.
Раздел 2. Простейшие программы. Реализация вычислений и ветвлений(12 ч.)					
3.	Типы данных в программировании. Определение переменной. Локальные и глобальные переменные.	4	4	8	Практические задания.
4.	Ввод данных с клавиатуры. Вывод данных на экран.	2	2	4	Практические задания.
5.	Логические выражения.	2	2	4	Практические задания.
6.	Условный оператор. Инструкция if.	2	2	4	Практические задания.
7.	Множественное ветвление.	2	2	4	Практические задания.
Раздел 3. Реализация циклических, вспомогательных алгоритмов. Рекурсия(21 ч.)					
8.	Цикл While.	2	2	4	Практические задания.
9.	Цикл For.	2	2	4	Практические задания.
10.	Вложенные циклы.	2	2	4	Практические задания.
11.	Процедуры.	4	4	8	Практические задания.
12.	Функции. Функции в программировании. Параметры и аргументы функций.	2	2	4	Практические задания.
13.	Рекурсия.	2	2	4	Практические задания.
14.	Сумма и произведение цифр числа.	2	2	4	Практические задания.
15.	Числа Фибоначчи (вычисление с помощью цикла while и рекурсии).	2	2	4	Практические задания.
16.	Алгоритм Евклида.	2	2	4	Практические задания.
17.	Итоговое занятие.	1	1	2	Тест и практические задания.
	ИТОГО:			72	

Учебно-тематический план 2 года обучения

№ п/п	Раздел (или тема) учебно-тематического плана	Количество часов			Формы контроля
		Теория	Практика	Всего	
1.	Вводное занятие. ТБ.	2	0	2	Беседа.
Раздел 1. Реализация циклических, вспомогательных алгоритмов. (10 ч.)					
2.	Вычисление факториала на языке программирования Python.	1	1	2	Практические задания.
3.	Перевод чисел в различные системы счисления.	2	2	4	Практические задания.
4.	Решето Эратосфена - алгоритм определения простых чисел.	1	1	2	Практические задания.
5.	Тестирование простоты числа методом перебора делителей.	1	1	2	Практические задания.
Раздел 2. Словари. Массивы. Обработка массивов (16 ч.)					
6.	Введение в словари.	1	1	2	Практические задания.
7.	Массивы. Основные задачи обработки массивов: поиск, реверс. Отбор элементов массива по условию.	3	3	6	Практические задания.
8.	Сортировка выбором (поиск минимума и перестановка).	1	1	2	Практические задания.
9.	Сортировка методом пузырька.	1	1	2	Практические задания.
10.	Двоичный (бинарный) поиск элемента в массиве. Замена элементов в списке.	2	2	4	Практические задания.
Раздел 3. Символьные строки. Обработка символьных строк. (10 ч.)					
11.	Строки как последовательности символов.	1	1	2	Практические задания.
12.	Функции для работы с символьными строками.	1	1	2	Практические задания.
13.	Преобразования «строка-число».	1	1	2	Практические задания.
14.	Строки в процедурах и функциях. Сравнение и сортировка строк.	2	2	4	Практические задания.
Раздел 4. Матрицы. Ввод, вывод, обработка матриц. Чтение и запись текстовых файлов. (16 ч.)					
15.	Матрицы. Ввод матриц с клавиатуры, с помощью генератора случайных чисел. Обработка матриц.	2	2	4	Практические задания.
16.	Файлы. Чтение текстового файла.	3	3	6	Практические задания.
17.	Запись в файл.	2	2	4	Практические задания.
Раздел 5. Разработка графического интерфейса программы. (18 ч.)					
18.	Графический пользовательский интерфейс. Основные виджеты.	1	1	2	Практические задания.
19.	Создание и размещение виджетов.	2	2	4	Практические

					задания.
20.	Доступ к значениям в виджетах.	1	1	2	Практические задания.
21.	Изменение конфигурации виджетов.	2	2	4	Практические задания.
22.	Запуск работы виджетов.	2	2	4	Практические задания.
23.	Итоговое занятие.	0	2	2	Тест и практические задания.
	ИТОГО:			72	

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 516
НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА

УТВЕРЖДЕН

Директор

Л.В. Смирнова

Приказ № 84-у от 27.05.2024 г.

**Календарный учебный график реализации дополнительной общеобразовательной
общеразвивающей программы «Основы программирования Python»**

Педагог: Бутко Маргарита Михайловна

Год обучения	Дата начала обучения программе	Дата окончания обучения программе	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год	01.09.2024	25.05.2025	36	36	72	1 раз в неделю по 1 часу
2 год	01.09.2024	25.05.2025	36	36	72	1 раз в неделю по 2 часа

РАБОЧАЯ ПРОГРАММА К ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЕ

«Основы программирования Python» 1 года обучения

Задачи:

Обучающие:

- познакомить с возможностями и особенностями современного языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных;
- научить применять функции при написании программ на языке программирования Python;
- сформировать систему знаний, умений и навыков, необходимых в работе программиста;
- обучить коммуникативным навыкам при совместной работе над проектом.

Развивающие:

- развить познавательные потребности и способности школьников;
- развить творческие способности, логическое мышление;
- развить внимательность и наблюдательность, привить навыки аккуратности и точности в работе;
- замотивировать к постоянному самообразованию.

Воспитательные:

- воспитать навыки взаимодействия при командной работе над проектом;
- воспитать чувство ответственности за результаты деятельности;
- сформировать осознанное отношение к выбору будущей профессии.

Планируемые результаты:

Предметные:

- будут ознакомлены с возможностями и особенностями современного языка программирования Python;
- будут изучены основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных;
- будут обучены применять функции при написании программ на языке программирования Python;
- будет сформирована система знаний, умений и навыков, необходимых в работе программиста;

Личностные:

- будут воспитаны навыки взаимодействия при командной работе над проектом;
- будет воспитано чувство ответственности за результаты деятельности;
- будет сформировано осознанное отношение к выбору будущей профессии.

Метапредметные:

- будут развиты познавательные потребности и способности школьников;
- будут развиты творческие способности, логическое мышление;
- будут развиты внимательность и наблюдательность, привиты навыки аккуратности и точности в работе;
- будут замотивированы к постоянному самообразованию.

Содержание программы 1 года обучения

Тема 1. Вводное занятие. Техника безопасности

Теория. Инструктаж учащихся по правилам поведения, технике безопасности работы в компьютерной лаборатории и правилам дорожного движения.

Раздел 1. Введение в курс

Тема 2. История языков программирования. Язык Python.

Теория. История языков программирования. Компиляция и интерпретация. Знакомство с Python и средами программирования.

Раздел 2. Простейшие программы. Реализация вычислений и ветвлений

Тема 3. Типы данных в программировании. Определение переменной. Локальные и глобальные переменные. Примеры программ.

Теория. Понятие типов данных. Основные типы данных. Понятие переменных в Python. Глобальные переменные. Локальные переменные.

Практика. Придумать два списка: допустимых и недопустимых имен переменных. Объяснить, почему каждое из имен допустимо или соответственно недопустимо. Затем придумать еще два списка — «хороших» и «плохих» допустимых имен — и объясните свой выбор. Пройти тест на проверку знаний по пройденной теме.

Тема 4. Ввод данных с клавиатуры. Вывод данных на экран. Примеры программ.

Теория. Функция input(). Функция print(). Форматирование строк. Операции над строками.

Практика. Написать программу, которая запрашивает у пользователя два числа и выводит на экран их сумму, разность, произведение и частное.

Написать программу, которая запрашивает у пользователя его имя, возраст, место жительства и выводит данные на экран.

Написать программу, которая запрашивает у пользователя 4 числа, затем складывает первые два и вторые два и делит первую сумму на вторую, выводит на экран результат в виде целого числа.

Тема 5. Логические выражения.

Теория. Понятие логических выражений. Логические выражения. Сложные логические выражения. Примеры программ.

Практика. Присвоить двум переменным любые числовые значения, с помощью операторов and и or составить два сложных логических выражения, одно из которых дает истину, второе – ложь.

Написать программу, которая запрашивает у пользователя два числа и выводит истину или ложь в зависимости от того, больше первое число второго или нет.

Тема 6. Условный оператор

Теория. Понятие ветвления. Условный оператор. Инструкция if. Примеры программ.

Практика. Написать программу, которая запрашивает у пользователя число и сообщает является ли оно положительным или отрицательным.

Написать программу, которая запрашивает у пользователя его возраст и стаж работы, а затем выводит сообщение, подходит ли тот в фирму, у которой есть требования: старше 20 лет и стаж больше 3 лет.

Тема 7. Множественное ветвление.

Теория. Расширенная инструкция if-else. Инструкция elif. Примеры программ.

Практика. Написать программу, которая запрашивает у пользователя число. Если оно, положительное, то на экран выводится цифра 1, если отрицательное, то -1. Если 0, то 0. Использовать в коде условный оператор множественного ветвления.

Раздел 3. Реализация циклических, вспомогательных алгоритмов. Рекурсия.

Тема 8. Цикл While.

Теория. Понятие цикла. Цикл While. Синтаксис цикла. Блок-схема цикла. Управление бесконечным циклом While. Пропуск итераций в цикле While. Примеры программ.

Практика. Написать программу, которая принимает на вход целые числа и вычисляет их сумму, пока пользователь не введет 0 (числа вводятся с клавиатуры).

Написать программу, которая поучает два числа и находит наименьшее число, которое без остатка делится на оба введенных числа.

Тема 9. Цикл For.

Теория. Цикл For. Синтаксис цикла. Блок-схема цикла. Функция range(). Цикл For с блоком else.

Практика. Написать программу, которая получает от пользователя строку целых чисел и выводит на экран: количество положительных чисел, произведение всех отрицательных чисел, минимальное и максимальное числа.

Тема 10. Вложенные циклы.

Теория. Понятие вложенных циклов. Синтаксис. Примеры программ.

Практика. Написать программу, которая генерирует таблицу умножения. Вводится одно натуральное число – требуемый размер таблицы.

Тема 11. Функции. Функции в программировании. Параметры и аргументы функций.

Теория. Понятие функции. Параметры функций. Значение аргументов функции по умолчанию. Примеры программ.

Практика. Написать функцию sum_range(start, end), которая суммирует все целые числа от значения «start» до величины «end» включительно. Если пользователь задаст первое число большее чем второе, просто поменять их местами.

Создайте функцию three_args(), которая принимает 1, 2 или 3 строго ключевых параметра. В результате ее работы на печать в консоль выводятся значения переданных переменных, но только если они не равны None.

Тема 12. Процедуры.

Теория. Функции в роли процедуры. Синтаксис функции-процедуры. Процедуры с параметрами. Примеры программ.

Практика. Создать процедуру, которая вычисляет разность 2-х вводимых пользователем чисел. Выполнить задание двумя способами: 1) процедура без параметров: числа – глобальные переменные, определенные в основной программе; 2) процедура с параметрами: числа – параметры процедуры.

Тема 13. Рекурсия.

Теория. Рекурсивны функции. Факториалы. Детали работы рекурсивной функции. Примеры программ.

Практика. Написать рекурсивную функцию sum_sub(list), которая будет принимать список целых чисел. Эта функция будет суммировать все нечётные числа и вычитать все чётные числа. В конце она будет возвращать получившееся значение.

Тема 14. Сумма и произведение цифр числа.

Теория. Сумма цифр числа. Произведение цифр числа. Частные случаи. Примеры программ.

Практика. Дано число. Найти сумму и произведение его цифр.

Вводится натуральное число. Найти сумму и произведение цифр, из которых состоит это число. При этом если в числе встречается цифра 0, то ее не надо учитывать при нахождении произведения.

Тема 15. Числа Фибоначчи.

Теория. Понятие чисел Фибоначчи. Вычисление n-го числа ряда Фибоначчи с помощью цикла while. Вывод чисел Фибоначчи циклом for. Рекурсивное вычисление n-го числа ряда Фибоначчи.

Практика. Написать программы всех описанных в теории способов.

Тема 16. Алгоритм Евклида.

Теория. Классический алгоритм Евклида. Расширенный алгоритм Евклида. Бинарный алгоритм Евклида.

Практика. Найти наибольший общий делитель для двух натуральных чисел с помощью алгоритма Евклида.

Тема 17. Итоговое занятие.

Практика. Прохождение итогового теста с практическими заданиями.

Календарно-тематическое планирование 1 года обучения

№ п/п	Раздел (или тема) учебно-тематического плана	Кол-во часов		Дата проведения		Форма контроля
		теория	практика	факт	план	
1	Вводное занятие. Техника безопасности.	1	1			Беседа
2-3	История языков программирования. Язык Python.	2	2			Тестирование
4-7	Типы данных в программировании. Определение переменной. Локальные и глобальные переменные.	4	4			Практическая работа
8-9	Ввод данных с клавиатуры. Вывод данных на экран.	2	2			Практическая работа
10-11	Логические выражения.	2	2			Практическая работа
12-13	Условный оператор.	2	2			Практическая работа
14-15	Множественное ветвление.	2	2			Практическая работа
16-17	Цикл While.	2	2			Практическая работа
18-19	Цикл For.	2	2			Практическая работа
20-21	Вложенные циклы.	2	2			Практическая работа
22-23	Функции. Функции в программировании. Параметры и аргументы функций.	2	2			Практическая работа
24-27	Процедуры.	4	4			Практическая работа
28-29	Рекурсия.	2	2			Практическая работа
30-31	Сумма и произведение цифр числа.	2	2			Практическая работа
32-33	Числа Фибоначчи (вычисление с помощью цикла while и рекурсии).	2	2			Практическая работа
34-35	Алгоритм Евклида.	2	2			Практическая работа
36	Итоговое занятие.	1	1			Тест и практическая работа
						72 часа

РАБОЧАЯ ПРОГРАММА К ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЕ «Основы программирования Python» 2 года обучения

Задачи:

Обучающие:

- познакомить с возможностями и особенностями современного языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных;
- научить применять функции при написании программ на языке программирования Python;
- сформировать систему знаний, умений и навыков, необходимых в работе программиста;
- обучить коммуникативным навыкам при совместной работе над проектом.

Развивающие:

- развить познавательные потребности и способности школьников;
- развить творческие способности, логическое мышление;
- развить внимательность и наблюдательность, привить навыки аккуратности и точности в работе;
- замотивировать к постоянному самообразованию.

Воспитательные:

- воспитать навыки взаимодействия при командной работе над проектом;
- воспитать чувство ответственности за результаты деятельности;
- сформировать осознанное отношение к выбору будущей профессии.

Планируемые результаты:

Предметные:

- будут ознакомлены с возможностями и особенностями современного языка программирования Python;
- будут изучены основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных;
- будут обучены применять функции при написании программ на языке программирования Python;
- будет сформирована система знаний, умений и навыков, необходимых в работе программиста;

Личностные:

- будут воспитаны навыки взаимодействия при командной работе над проектом;
- будет воспитано чувство ответственности за результаты деятельности;
- будет сформировано осознанное отношение к выбору будущей профессии.

Метапредметные:

- будут развиты познавательные потребности и способности школьников;
- будут развиты творческие способности, логическое мышление;
- будут развиты внимательность и наблюдательность, привиты навыки аккуратности и точности в работе;
- будут замотивированы к постоянному самообразованию.

Содержание программы 2 года обучения

Тема 1. Вводное занятие. Техника безопасности

Теория. Инструктаж учащихся по правилам поведения, технике безопасности работы в компьютерной лаборатории и правилам дорожного движения.

Раздел 1. Реализация циклических, вспомогательных алгоритмов.

Тема 2. Вычисление факториала на языке программирования Python.

Теория. Понятие факториалов. Вычисление факториала на языке программирования Python.

Практика. Написать рекурсивную функцию `factorial(n)`, которая будет принимать положительное целое число `n` и возвращать факториал от этого числа ($1 \times 2 \times 3 \times \dots \times n$).

Тема 3. Перевод чисел в различные системы счисления.

Теория. Понятие систем исчисления. Инstrukция `bin()`. Инstrukция `hex()`. Инstrukция `oct()`. Преобразование вручную.

Практика. Написать программу, которая переводит из десятичной системы в двоичную, восьмеричную, шестнадцатеричную системы. Число вводится с клавиатуры.

Тема 4. Решето Эратосфена - алгоритм определения простых чисел.

Теория. Понятие простых чисел. Решето Эратосфена. Блок-схема программы. Пример программы.

Практика. Написать программу с решето Эратосфена, где граница верхнего диапазона вводится с клавиатуры.

Тема 5. Тестирование простоты числа методом перебора делителей.

Теория. Перебор делителей. Блок-схема тестирования числа на простоту. Пример программы.

Практика. Написать программу, которая находит все простые числа и выводит на экран их сумму и произведение.

Раздел 2. Словари. Массивы. Обработка массивов.

Тема 6. Введение в словари.

Теория. Понятие словарей. Получение данных из словаря. Ключи в словарях. Методы. Итерация через словарь.

Практика. Написать функцию `to_dict(lst)`, которая принимает аргумент в виде списка и возвращает словарь, в котором каждый элемент списка является и ключом, и значением. Предполагается, что элементы списка будут соответствовать правилам задания ключей в словарях.

Иван решил создать самый большой словарь в мире. Для этого он придумал функцию `biggest_dict(**kwargs)`, которая принимает неограниченное количество параметров «ключ: значение» и обновляет созданный им словарь `my_dict`, состоящий всего из одного элемента «`first_one`» со значением «`we can do it`». Воссоздайте эту функцию.

Тема 7. Массивы. Основные задачи обработки массивов: поиск, реверс. Отбор элементов массива по условию.

Теория. Понятие массивов. Синтаксис программы. Поиск. Реверс. Отбор элементов массива по условию.

Практика. Написать программу, которая находит наибольшее число в одномерном массиве. Размер массива вводится с клавиатуры. Элементы заполняются рандомно.

Написать программу, которая меняет местами первый и последний элементы массива. Размер массива вводится с клавиатуры. Элементы заполняются рандомно.

Тема 8. Сортировка выбором (поиск минимума и перестановка).

Теория. Алгоритм сортировки выбором. Блок-схема сортировки выбором. Пример программы.

Практика. Написать программу, которая сортирует выбором нечетные элементы массива.

Тема 9. Сортировка методом пузырька.

Теория. Алгоритм сортировки методом пузырька. Блок-схема. Пример программы.

Практика. Дан список целых чисел, состоящий из 30 элементов. Найти сумму пяти самых больших и пяти самых маленьких элементов списка.

Тема 10. Двоичный (бинарный) поиск элемента в массиве. Замена элементов в списке.

Теория. Алгоритм двоичного поиска элемента в массиве. Блок-схема. Создание списков. Замена элементов в списке. Примеры программ.

Практика. Написать программу, которая возвращает номер элемента в массиве, который равен вашему возрасту. Если такого нет, выводить сообщение об отсутствии.

Написать программу, которая заменяет в списке все цифры 1 на 0.

Раздел 3. Символьные строки. Обработка символьных строк.

Тема 11. Строки как последовательности символов.

Теория. Понятие строк. Создание строк. Экранирование строк. Методы для работы со строками. Примеры программ.

Практика. Напишите функцию `search_substr(subst, st)`, которая принимает 2 строки и определяет, имеется ли подстрока `subst` в строке `st`. В случае нахождения подстроки, возвращается фраза «Есть контакт!», а иначе «Мимо!». Должно быть найдено совпадение независимо от регистра обеих строк.

Требуется определить индексы первого и последнего вхождения буквы в строке. Для этого нужно написать функцию `first_last(letter, st)`, включающую 2 параметра: `letter` – искомый символ, `st` – целевая строка. В случае отсутствия буквы в строке, нужно вернуть кортеж (`None, None`), если же она есть, то кортеж будет состоять из первого и последнего индекса этого символа.

Тема 12. Функции для работы с символьными строками.

Теория. Понятие символьных строк. Сравнение строк. Арифметические операции. Обращение к символам. Встроенные методы.

Практика. Вводится строка, в которой сначала записана фамилия человека, а затем через пробел – его имя, например, "Семёнов Андрей". Записать операторы, которые позволяют:

- найти номер пробела, разделяющего фамилию и имя, и записать его в переменную `p`;
- выделить из строки фамилию и записать её в переменную `fam`;
- выделить из строки имя и записать его в переменную `name`;
- приписать перед фамилией первую букву имени, точку и пробел.

Тема 13. Преобразования «строка-число».

Теория. Преобразования числа в строку. Преобразование строки в число. Примеры программ.

Практика. Какие из этих строк можно преобразовать в целое число, какие – в вещественное: "45", "5 p.", "14.5", "14;5", "tu154", "543.0", "(30)", "1E3".

Тема 14. Строки в процедурах и функциях. Сравнение и сортировка строк.

Теория. Использование строк в процедурах и функциях. Алгоритм сортировки символов в строке. Сравнение строк.

Практика. Написать программу, которая сортирует элементы строки по возрастанию и убыванию. Значение строки вводится с клавиатуры.

Раздел 4. Матрицы. Ввод, вывод, обработка матриц. Чтение и запись текстовых файлов.

Тема 15. Матрицы. Ввод матриц с клавиатуры, с помощью генератора случайных чисел. Обработка матриц.

Теория. Понятие матрицы. Создание матрицы. Ввод и вывод матрицы. Генератор случайных чисел. Обработка матрицы. Примеры программ.

Практика. Написать программу поиска минимального и максимального элементов матрицы и их индексов.

Написать программу, выводящую на экран строку матрицы, сумма элементов которой максимальна.

Тема 16. Файлы. Чтение текстового файла.

Теория. Файлы в Python. Текстовые и бинарные файлы. Открытие файла. Закрытие файла. Чтение файла. Примеры программ.

Практика. Документ «article.txt» содержит следующий текст:

Вечерело

Жужжали мухи

Светил фонарик

Кипела вода в чайнике

Венера зажглась на небе

Деревья шумели

Тучи разошлись

Листва зеленела

Написать функцию `read_last(lines, file)`, которая будет открывать определенный файл `file` и выводить на печать построчно последние строки в количестве `lines` (на всякий случай проверим, что задано положительное целое число). Протестировать функцию на файле «article.txt»

Требуется реализовать функцию `longest_words(file)`, которая выводит слово, имеющее максимальную длину (или список слов, если таковых несколько).

Тема 17. Запись в файл.

Теория. Режимы доступа к файлам. Функции записи в файл. Примеры программ.

Практика. Пользователь вводит предложение с клавиатуры. Разбейте это предложение по словам (считать, что слова разделены пробелом) и сохраните их в столбец в файл.

Пусть имеется словарь: `d = {"house": "дом", "car": "машина", "tree": "дерево", "road": "дорога", "river": "река"}`. Необходимо каждый элемент этого словаря сохранить в бинарном файле как объект. Затем, прочитать этот файл и вывести считанные объекты в консоль.

Раздел 5. Разработка графического интерфейса программы.

Тема 18. Графический пользовательский интерфейс. Основные виджеты.

Теория. Понятие графического пользовательского интерфейса. Понятие виджетов. Основные виджеты.

Практика. Потренироваться создавать различные виджеты.

Тема 19. Создание и размещение виджетов.

Теория. Алгоритм создания виджетов. Особенности создания. Демонстрация создания. Размещение виджетов в окне. Метод `pack()`.

Практика. Начать разработку пользовательского интерфейса в соответствии с выбранным вариантом.

Тема 20. Доступ к значениям в виджетах.

Теория. Использование значений виджетов. Особенности использования значений виджетов.

Практика. Продолжить разработку пользовательского интерфейса в соответствии с выбранным вариантом.

Тема 21. Изменение конфигурации виджетов.

Теория. Метод `config()`. Метод `insert()`.

Практика. Продолжить разработку пользовательского интерфейса в соответствии с выбранным вариантом.

Тема 22. Запуск работы виджетов.

Теория. Добавление функционала виджетам. Методы для добавления. Пример программы.

Практика. Продолжить разработку пользовательского интерфейса в соответствии с выбранным вариантом.

Тема 23. Итоговое занятие.

Практика. Прохождение итогового теста с практическими заданиями и демонстрация своего пользовательского интерфейса.

Календарно-тематическое планирование 2 года обучения

№ п/п	Раздел (или тема) учебно-тематического плана	Кол-во часов		Дата проведения		Форма контроля
		теория	практика	факт	план	
1	Вводное занятие. Техника безопасности.	2	0			Беседа.
2	Вычисление факториала на языке программирования Python.	1	1			Практическая работа
3	Перевод чисел в различные системы счисления.	1	1			Практическая работа
4	Решето Эратосфена - алгоритм определения простых чисел.	1	1			Практическая работа
5	Тестирование простоты числа методом перебора делителей.	1	1			Практическая работа
6-7	Введение в словари.	2	2			Практическая работа
8-9	Массивы. Основные задачи обработки массивов: поиск, сортировка, реверс. Отбор элементов массива по условию.	2	2			Практическая работа
10	Сортировка выбором (поиск минимума и перестановка).	1	1			Практическая работа
11	Сортировка методом пузырька.	1	1			Практическая работа
12-13	Двоичный (бинарный) поиск элемента в массиве. Отбор элементов массива по условию. Замена элементов в списке.	2	2			Практическая работа
14-15	Строки как последовательности символов.	2	2			Практическая работа
16-17	Функции для работы с символьными строками.	2	2			Практическая работа
18	Преобразования «строка-число».	1	1			Практическая работа
19-20	Строки в процедурах и функциях. Сравнение и сортировка строк.	2	2			Практическая работа
21-22	Матрицы. Ввод матриц с клавиатуры, с помощью генератора случайных чисел. Обработка матриц.	2	2			Практическая работа

23-24	Файлы. Чтение текстового файла.	2	2			Практическая работа
25	Запись в файл.	1	1			Практическая работа
26-27	Графический пользовательский интерфейс. Основные виджеты.	2	2			Практическая работа
28-29	Создание и размещение виджетов.	2	2			Практическая работа
30-31	Доступ к значениям в виджетах.	2	2			Практическая работа
32-33	Изменение конфигурации виджетов.	2	2			Практическая работа
34-35	Запуск работы виджетов.	2	2			Практическая работа
36	Итоговое занятие.	0	2			Практическая работа
				72 часа		

Методические и оценочные материалы
Методические материалы.

№	Раздел программы	Форма организации занятия	Методы и приемы	Дидактический материал, техническое оснащение	Формы подведения итогов
1	Введение в курс.	Лекция	Лекция	Лекционный материал	Тестирование
2	Простейшие программы. Реализация вычислений и ветвлений	Лекция, практическое занятие	Лекция, практическая работа	Тематические текстовые подборки, демонстрация программы по теме с помощью проектора	Практическая работа
3	Реализация циклических, вспомогательных алгоритмов. Рекурсия	Лекция, практическое занятие	Лекция, практическая работа	Тематические текстовые подборки, демонстрация программы по теме с помощью проектора	Практическая работа
4	Словари. Массивы. Обработка массивов.	Лекция, практическое занятие	Лекция, практическая работа	Тематические текстовые подборки, демонстрация программы по теме с помощью проектора	Практическая работа
5	Символьные строки. Обработка символьных строк	Лекция, практическое занятие	Лекция, практическая работа	Тематические текстовые подборки, демонстрация программы по теме с помощью проектора	Практическая работа
6	Матрицы. Ввод, вывод, обработка матриц. Чтение и запись текстовых файлов.	Лекция, практическое занятие	Лекция, практическая работа	Тематические текстовые подборки, демонстрация программы по теме с помощью проектора	Практическая работа
7	Разработка графического интерфейса программы.	Лекция, практическое занятие	Лекция, практическая работа	Тематические текстовые подборки, демонстрация программы по теме с помощью проектора	Практическая работа

Информационные источники

Литература для учителя:

1. Домашняя страница Python www.python.org . Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
4. Сайт проекта Open Book Project openbookproject.net содержит серию практических примеров на Python Криса Мейерса.
5. *Python. Подробный справочник* Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
6. *Python. Справочник* Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.

Литература для учащегося:

1. Домашняя страница Python www.python.org . Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
4. Сайт проекта Open Book Project openbookproject.net содержит серию практических примеров на Python Криса Мейерса.
5. *Python. Подробный справочник* Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
1. *Python. Справочник* Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.

Оценочные материалы

Для отслеживания результативности образовательной деятельности по программе проводятся: входной, текущий контроль, промежуточная аттестация, итоговое оценивание.

Входной контроль проводится в начале курса с целью выявления первоначального уровня знаний и умений, возможностей детей.

Текущий контроль осуществляется на занятиях в течение всего курса для отслеживания уровня освоения учебного материала программы. Формы:

- педагогическое наблюдение;
- выполнение тестовых заданий на знание терминологии 3D-моделирования;
- практические задания.

Промежуточная аттестация предусмотрена 2 раза в год (декабрь, май) с целью выявления уровня освоения программы учащимися и уровня развития личностных качеств. Формы:

- устный и письменный опрос;
- выполнение тестовых и практических заданий по темам программы.

Итоговое оценивание проводится в конце обучения по программе. Формы:

- презентация продукта;
- анкетирование детей и родителей с целью выявления степени удовлетворенности образовательным процессом в коллективе и учреждении.

Карта оценки результативности учащегося по дополнительной общеразвивающей программе

№ п/п	Результаты	Параметры оценки уровня освоения программы	Характеристика низкого уровня освоения программы	Оценка уровня освоения программы (в баллах)					Характеристика высокого уровня освоения программы
				Очень слабо	Слабо	Удовлетворительно	Хорошо	Очень хорошо	
1	Предметные результаты	Опыт освоения теоретической информации (объём, прочность, глубина)	Информация не освоена	1	2	3	4	5	Информация освоена полностью в соответствии с задачами программы
2	Предметные результаты	Опыт практической деятельности (степень освоения способов деятельности: умения и навыки)	Способы деятельности не освоены	1	2	3	4	5	Способы деятельности освоены полностью в соответствии с задачами программы
3	Личностные результаты	Опыт самопрезентации	Отсутствует опыт самопрезентации (неуверенная речь, в презентации отсутствуют основные моменты)	1	2	3	4	5	Приобретён отличный опыт самопрезентации (во время презентации обучающийся хорошо продемонстрировал себя и свой продукт)
4	Личностные результаты	Опыт общения и поведения в социуме	Общение отсутствовало (ребёнок закрыт для общения)	1	2	3	4	5	Приобретён опыт взаимодействия и сотрудничества в системах «педагог-учащийся» и «учащийся-учащийся».
5	Метапредметные результаты	Опыт творчества	Освоены элементы репродуктивной, имитационной деятельности	1	2	3	4	5	Приобретён опыт самостоятельной творческой деятельности (оригинальность, индивидуальность, качественная завершенность результата)
			Итоговый балл						-

Общая оценка уровня освоения программы:

5-14 баллов – программа освоена на низком уровне;

15-19 баллов – программа освоена на среднем уровне;

20-25 баллов – программа освоена на высоком уровне.

Ведомость итогового контроля результативности учащихся по дополнительной общеразвивающей программе

Код группы: _____ **учебный период:** _____ **ФИО педагога:** _____

Название программы: _____

№ п/п	ФИО учащегося	Оценка уровня освоения программы учащимися						Итоговый балл	Уровень освоения программы (низкий, средний, высокий)
		Опыт освоения теоретической информации (объём, прочность, глубина)	Опыт практической деятельности (степень освоения способов деятельности: умения и навыки)	Опыт эмоционально-ценностных отношений (вклад в формирование личностных качеств учащегося)	Опыт общения и поведения в социуме	Опыт творчества	Мотивация и осознание перспективы		
1.									
2.									
3.									
4.									
5.									
6.									
7.									
8.									
9.									
10.									
11.									
12.									
13.									
14.									
15.									

_____ Дата

_____ Подпись

_____ ФИО педагога