

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Министерство образования и молодежной политики Свердловской области
Муниципальное казенное учреждение "Управление образования
администрации городского округа "Город Лесной"
МБОУ СОШ № 75

УТВЕРЖДЕНО

Директор



Могиленских Т.А.

23 08 2024 г.

ДОПОЛНИТЕЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

«Программирование (PYTHON)»

Срок реализации - 1 год

Составитель программы:

Солодухин Евгений Анатольевич

педагог дополнительного образования

Лесной, 2024

Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование (PYTHON)» имеет техническую направленность. В современную жизнь человека все шире внедряются компьютеры и информационные технологии. Поэтому все большее значение приобретает компьютерная грамотность.

Новизна программы заключается в том, что Python дает широкие возможности в области программирования. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Актуальность программы

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с легкостью научиться программировать. Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Еще одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Курс разработан для получения первичных знаний в области программирования на языке Python обучающихся 11 - 16 лет

Цель программы: обучение учащихся программированию посредством языка Python, развитие инженерного мышления, воспитание конкурентоспособной личности.

Задачи программы

Обучающие:

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;

- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

Развивающие:

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

Воспитательные:

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

Программа рассчитана на 1 год по 4 часа в неделю (70 часов) обучения.

Возраст обучающихся - с 11 до 16 лет.

Продолжительность 2 часа (по 45 минут) 2 раза в неделю

Количество обучающихся 10 человек.

Планируемые результаты

Личностные результаты:

- формирование умения самостоятельной деятельности;
- формирование умения работать в команде;
- формирование коммуникативных навыков;
- формирование навыков анализа и самоанализа;
- формирование целеустремлённости и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

Предметные результаты:

- формирование понятий «алгоритм», «программа»;
- формирование понятий об основных конструкциях языка программирования Python, таких как оператор ветвления if, операторы цикла while, for, вспомогательные алгоритмы;
- формирование понятий о структурах данных языка программирования Python;
- формирование основных приёмов составления программ на языке программирования Python;

- формирование алгоритмического и логического стилей мышления.

Метапредметные результаты:

- формирование умения ориентироваться в системе знаний;
- формирование умения выбирать наиболее эффективные способы решения задач на компьютере в зависимости от конкретных условий;
- формирование приёмов проектной деятельности, включая умения видеть проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, результат деятельности соотносить с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы;
- формирование умения распределять время; • формирование умений успешной самопрезентации.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование раздела, темы	Количество часов			Форма аттестации/ контроля
		Всего	Теория	Практика	
1	Введение в Python	14	6	8	
1.1	Язык Python. Среда программирования. Особенности ввода/вывода	4	2	2	Практическое задание
1.2	Типы данных, операции. Оператор присваивания.	4	2	2	Выполнение упражнений
1.3	Числа. Стандартные операции.	6	2	4	Практическое задание
2	Алгоритмические инструкции	14	6	8	
2.1	Условный оператор	4	2	2	Выполнение упражнений

2.2	Цикл while. Теория чисел	5	2	3	Практическое задание
2.3	Цикл for	5	2	3	Практическое задание
3	Строки	14	5	9	
3.1	Литералы строк	3	1	2	Практическое задание
3.2	Срезы строк	3	1	2	Практическое задание
3.3	Методы строк	8	3	5	Практическое задание
4	Функции	12	6	6	
4.1	Парадигма и преимущества структурного программирования	2	2	-	Опрос
4.2	Граф и стек вызовов функций. Области видимости переменных	3	1	2	Практическое задание
4.3	Прямая рекурсия	4	2	2	Практическое задание
4.4	Косвенная рекурсия	3	1	2	Практическое задание
5	Списки и кортежи	12	6	6	
5.1	Списки и кортежи в Python. Сходства и различия	1	1	-	Опрос

5.2	Операции со списками	3	1	2	Практическое задание
5.3	Срезы списков	4	2	2	Практическое задание
5.4	Матрицы. Операции над матрицами	4	2	2	Практическое задание
6.	Заключительное занятие	4	-	4	
6.1	Подведение итогов. Индивидуальный проект.	4		4	Представление программ
	ВСЕГО	70	29	41	

СОДЕРЖАНИЕ УЧЕБНОГО ПРЕДМЕТА

1. Введение в Python

Основные понятия: трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная,

динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты

операций, литералы чисел, операция присваивания, PEP 8.

1.1. Язык Python. Среда программирования. Особенности ввода/вывода

Теория. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы.

Практика. Установка и настройка IDLE разных версий.

1.2. Типы данных, операции. Оператор присваивания

Теория. Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

1.3. Числа. Стандартные операции

Теория. Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).

Практика. Решение простых задач в интерактивном режиме.

2. Алгоритмические инструкции

Основные понятия: логический тип данных, логические операции (and, or, not, A), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла,

переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

2.1. Условный оператор

Теория. Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

Практика. Решение задач.

2.2. Цикл while

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики.

2.3. Цикл for

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

3. Строки

Основные понятия: символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape- последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

3.1. Литералы строк

Теория. Понятие «литералы строк».

Практика. Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

3.2. Срезы строк

Теория. Форматирование строки.

Практика. Преобразование строки. Решение упражнений.

3.3. Методы строк

Теория. Методы работы со строкой.

Практика. Применение методов строки. Решение задач.

4. Функции

Основные понятия: подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

4.1. Парадигма и преимущества структурного программирования

Теория. Обзор парадигм программирования. Особенности применения языков программирования.

4.2. Граф и стек вызовов функций. Области видимости переменных

Теория. Применение стек и граф вызовов функций.

Практика. Выполнение тренировочных упражнений.

4.3. Прямая рекурсия

Теория. Понятие прямой рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

4.4. Косвенная рекурсия

Теория. Понятие косвенной рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений. Практические занятия к темам 4.2., 4.3., 4.4.: Нахождение суммы чисел. Числа Фибоначчи. Вычисление степени. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности.

5. Списки и кортежи

Основные понятия: список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировка списка, случайное перемешивание.

5.1. Списки и кортежи в Python. Сходства и различия

Теория. Представление списка и кортежа в памяти компьютера, сходства и различия.

5.2. Операции со списками

Теория. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка.

5.3. Срезы списков

Теория. Методы работы со списком и кортежем. Методы сортировки списка.

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

5.4. Матрицы. Операции над матрицами

Теория. Вычисление сложности алгоритма. Многомерные списки.

Практика. Решение задач повышенной трудности.

8. Подведение итогов

8.1. Защита индивидуального проекта.

Представление разработанных программ по темам.

Список литературы

1. Доусен М. Програмуем на Python / М. Доусен - СПб.: Питер, 2016. - 416с.
2. Лутц М. Изучаем Python, 4 издание / М. Лутц - СПб.: Символ-Плюс, 2011. - 1280 с.
3. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016. - 480с.
4. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений / Н.А. Прохоренок, В.А. Дронов - СПб.: «БХВ- Петербург», 2016. - 832с.
5. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014. - 338с.