ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ САМАРСКОЙ ОБЛАСТИ ОСНОВНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА ПОС. КОШЕЛЕВКА МУНИЦИПАЛЬНОГО РАЙОНА СЫЗРАНСКИЙ САМАРСКОЙ ОБЛАСТИ

Рассмотрено и принято на заседании методического объединения учителей Протокол № 1 от 17.08.2021 г. Руководитель МО: _____Н.В.Марченко	Проверено. Рекомендовано к утверждению 17 августа 2021 г. Зам.директора по УВР: _____И.А.Рагушина	Утверждено к использованию в образовательном процессе Учреждения Приказ № 271 от 17.08.2021 г. Директор: _____Л.Е.Юсупова

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА технологической направленности

«PYTHON. Программирование для любознательных»

Уровень: базовый

Возраст обучающихся: 11 - 15 лет

Дополнительная общеобразовательная (общеразвивающая) программа **«PYTHON. Программирование для любознательных»** имеет **техническую** направленность.

Дополнительность программы заключается в том, что она отвечает интересам детей, способствует углублению знаний, умений и навыков, приобретаемых в общеобразовательной школе на уроках информатики.

Актуальность программы.

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью 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. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Отличительные особенности программы.

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

Новизна программы

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. 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 интернетресурсы такие, как Google, YouTube, также разработаны с

помощью языка программирования Python.

Педагогическая целесообразность Программы

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному».

Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

Адресат программы.

Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

Набор производится по желанию учащихся и их родителей.

Срок реализации программы составляет 1 год. Общее количество часов в год составляет 34 часа.

Форма обучения – очная форма реализации программы.

Особенности организации образовательного процесса - групповые – для всей группы при изучении общих практических и теоретических вопросов. Наполняемость группы до 15 человек. Состав группы постоянный.

В ходе реализации программы применяется дифференцированный, индивидуальный подход к каждому обучающемуся.

Занятия проводятся 1 раза в неделю по 1 часа (45 минут).

Уровень реализуемой программы – базовый.

Цель Программы: Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

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.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ ПРОГРАММЫ ОБЪЕДИНЕНИЯ

Личностные универсальные учебные действия

У учащихся будут сформированы:

- положительное отношение к исследовательской и экспериментальной деятельности;
- интерес к новому содержанию и новым способам познания;
- ориентация на понимание причин успеха в исследовательской и экспериментальной деятельности, на самоанализ и самоконтроль результата, на анализ соответствия результатов требованиям конкретной задачи;
- способность к самооценке на основе критериев успешности исследовательской и экспериментальной деятельности.

Учащиеся получат возможность для формирования:

- внутренней позиции учащегося на уровне понимания необходимости исследовательской и экспериментальной деятельности;
- познавательной мотивации;
- устойчивого интереса к новым способам познания;
- адекватного понимания причин успешности/неуспешности исследовательской и экспериментальной деятельности;
- морального сознания, способности к решению моральных проблем на основе учета позиций партнеров в общении, устойчивого следования в поведении моральным нормам и этическим требованиям.

Регулятивные универсальные учебные действия

Учащиеся научатся:

- принимать и сохранять учебную задачу;
- учитывать выделенные учителем ориентиры действия;
- планировать свои действия;
- осуществлять итоговый и пошаговый контроль;
- адекватно воспринимать оценку учителя;
- различать способ и результат действия;
- вносить коррективы в действия на основе их оценки и учета сделанных ошибок

Учащиеся получат возможность научиться:

- проявлять познавательную инициативу;
- самостоятельно учитывать выделенные учителем ориентиры действия в незнакомом материале;

- самостоятельно находить варианты решения познавательной задачи.

Познавательные универсальные учебные действия

Учащиеся научатся:

- осуществлять поиск нужной информации для выполнения учебного исследования с использованием учебной и дополнительной литературы в открытом информационном пространстве, в т.ч. контролируемом пространстве Интернет;
- использовать знаки, символы, модели, схемы для решения познавательных задач и представления их результатов;
- планировать и проводить наблюдения и эксперименты, высказывать суждения, делать умозаключения и выводы, аргументировать (защищать) свои идеи и т.п.

Учащиеся получат возможность научиться:

- осуществлять расширенный поиск информации в соответствии с исследовательской и экспериментальной задачей с использованием ресурсов библиотек и сети Интернет;
- фиксировать информацию с помощью инструментов ИКТ;
- осознанно и произвольно строить сообщения в устной и письменной форме;
- оперировать такими понятиями, как явление, причина, следствие, событие, обусловленность, зависимость, различие, сходство, общность, совместимость, несовместимость, возможность, невозможность и др.;

Коммуникативные универсальные учебные действия

Учащиеся научатся:

- допускать существование различных точек зрения;
- формулировать собственное мнение и позицию;
- договариваться, приходить к общему решению;
- соблюдать корректность в высказываниях;
- задавать вопросы по существу;
- использовать речь для регуляции своего действия;
- контролировать действия партнера;
- владеть монологической и диалогической формами речи.

Учащиеся получат возможность научиться:

- *аргументировать свою позицию и координировать ее с позицией партнеров при выработке общего решения в совместной деятельности;*
- *передавать партнеру необходимую информацию, с учетом целей коммуникации достаточно полно и точно*
- *осуществлять взаимный контроль и оказывать партнерам в сотрудничестве необходимую взаимопомощь;*

- *адекватно использовать речь для планирования и регуляции своей деятельности.*

1.2. Содержание программы.

Содержание учебного (тематического) плана

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 языка программирования Python 3.5 и среды программирования WinglDE 100. Регистрация на Интернет-ресурсах.

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.

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. Вычисление степени. Числа Фибоначчи. 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 сортировки 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.

6. *Словари и множества* Основные понятия: *словарь, множество, ключ, кодирование.*

6.1. Словари

Теория. Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

Практика. Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

6.2. Множества

Теория. Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

Практика. Решение задач повышенной трудности.

7. *Объектно-ориентированное программирование (ООП)*

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

7.1. Классы в Python

Теория. Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс»,

«событие».

7.2. Разработка собственного класса

Теория. Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События. ***Практика.*** Создание собственного класса.

7.3. Разработка и программирование собственного проекта

Практика. Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и

презентации проекта.

8. *Подведение итогов* 8.1. Защита индивидуального проекта.

КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№ п/п	Наименование раздела, темы	Количество часов			Используемое оборудование центра «Точка роста»
		Всего	Теория	Практика	
1.	**Введение в Python**	**3**	**2**	**1**	
1.1.	Язык Python. Среда программирования. Особенности ввода-вывода	1	1	1	Ноутбук, проектор, колонки Образовательный набор для изучения многокомпонентных робототехнических систем и манипуляционных роботов Комплект конструктивных элементов из металла и пластика для сборки моделей манипуляционных роботов с угловой кинематикой,
1.2.	Типы данных, операции. Оператор присваивания	1	1	-	
1.3.	Числа. Стандартные операции	1	-	-	
2.	**Алгоритмические конструкции**	8	4	4	
2.1.	Условный оператор	1	2	-	Образовательный набор для изучения многокомпонентных робототехнических систем и манипуляционных роботов Знакомство с оборудованием конструктора LEGO NXT Mindstorms EV3 электронные компоненты,
2.2.	Цикл while. Теория	5	2	3	
	чисел				
2.3.	Цикл for	2	-	1	
3.	**Строки**				
3.1.	Литералы строк	4	2	2	
3.2.	Срезы строк	4	2	2	

3.3.	Методы строк	12	4	8	соединительные и конструкционные элементы.
4.	**Функции**				
4.1.	Парадигма и преимущества структурного программирования	2	2	-	Образовательный набор для изучения многокомпонентных робототехнических систем и манипуляционных роботов Датчик скорости Кнопка тактовая Программируемый контроллер Шайба Нейлон Стойка 15 мм мамамама Силовая плата расширения Стойка 20 мм мамамама Винт DIN 7985 Полиамидная втулка Ф7-3х5 Винт ISO 4762 Винт М3х10 Плата расширения DXL-loT с адаптером Ethernet
4.2.	Граф и стек вызовов функций. Области видимости переменных	4	2	2	
4.3.	Прямая рекурсия	10	2	8	

4.4.	Косвенная рекурсия	4	2	2	Образовательный набор для изучения многокомпонентных робототехнических систем и манипуляционных роботов
5.	**Списки и кортежи**				

5.1.	Списки и кортежи в Python. Сходства и различия	2	2	-	Образовательный набор для изучения многокомпонентных робототехнических систем и манипуляционных роботов Кабель USB Гайка ISO 4035 Винт M3x8 Винт M3x6 Гайка M3 Гайка M3 Стойка 25 мм мамамама Гайка ISO 4035 Винт ISO 7046 -1 M2x6 Комплект конструктивных элементов
5.2.	Операции со списками	1	2	8	
5.3.	Срезы списков	4	2	2	
5.4.	Матрицы. Операции над матрицами.	4	2	2	
6	**Словари и множества**	**1**	**1**	**1**	
6.1.	Словари	1	1	1	Образовательный набор для изучения многокомпонентных робототехнических

6.2.	Множества	1	1	1	систем и манипуляционных роботов Ноутбук, проектор
7.	**Объектноориентированное программирование**	**6**	**3**	**4**	
7.1.	Классы в Python	2	2	-	Образовательный набор для изучения многокомпонентных робототехнических систем и манипуляционных роботов
7.2.	Разработка собственного класса	2	1	2	
7.3.	Разработка и	2	-	2	
	программирование собственного проекта				
8.	**Заключительное занятие**	**1**	**-**	**1**	
8.1.	Подведение итогов. Индивидуальный проект	1		1	Защита проекта
	Всего	**34**			

C=RU, O=ГБОУ ООШ пос. Кошелевка, CN=Людмила Евгеньевна Юсупова, E=koshsh@mail.ru
00 f6 47 9b 7f 10 11 36 2f
2022-07-10 16:09:40