

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа Базарно-Карабулакского
муниципального района Саратовской области»

Принята на заседании Педагогического совета Протокол № <u>1</u> от « <u>29</u> » <u>12</u> 2024г.	УТВЕРЖДАЮ Директор школы <u>[подпись]</u> /Тугушева А.Р./ Приказ № <u>129</u> от « <u>30</u> » <u>12</u> 2024 г
---	--



**Дополнительная общеобразовательная общеразвивающая
программа технической направленности
«Программирование: Python»**

*Возраст обучающихся – 12-17 лет,
Срок реализации: 1 год*

**Автор-составитель
программы:**

Дамаев Ряшит Харисович.
Педагог дополнительного
образования.

**с.Яковлевка
2024 год**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Программирование: Python» разработана в соответствии со следующими нормативно-правовыми документами:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ» с последующими дополнениями и изменениями;
- Приказ Министерства просвещения РФ от 09.11.2018 г. №196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- Санитарно-эпидемиологические требованиями к содержанию и организации режима работы образовательных организаций дополнительного образования детей (СанПиН 2.4.4. 3172-14»).

Направленность программы – техническая. Дополнительная общеобразовательная программа «Программирование: Python» является общеразвивающей программой технической направленности.

Дополнительная образовательная программа «Программирование: Python» является прикладной, носит практико-ориентировочный характер и направлена на овладение воспитанниками основных приемов программирования на языке программирования Python. Обучение по данной программе создает благоприятные условия для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

Актуальность программы. Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения выбран язык Python и обусловлено тем, что синтаксис языка достаточно прост и интуитивно понятен, что позволяет сосредоточиться на логических и алгоритмических аспектах программирования, не выучивая тонкости синтаксиса. Являясь очень востребованным языком, он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных так и прикладных задач, позволит потом с легкостью выучить любой другой язык программирования. Приобретенные знания и умения могут быть использованы обучающимися при сдаче ОГЭ и ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистики и другим наукам.

Педагогическая целесообразность программы заключается в том, что дети приобретут практические навыки, которые станут основой для дальнейшего изучения основ программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации обучающихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Осваивая данную программу, обучающиеся будут овладевать навыками, которые будут востребованы в ближайшие десятилетия в специальностях, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, получаемые в процессе обучения по программе.

Отличительная особенность программы состоит в том, что она позволяет привлечь детей среднего школьного возраста к изучению программирования при помощи языка Python, так как он обладает следующими достоинствами:

- Python - это текстовый язык программирования. Он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

- Python – простой и удобный язык по сравнению со многими другими языками.

- В Python есть библиотеки готовых процедур для использования в своих программах.

Это позволяет создавать сложные программы быстро;

Назначение программы:

Данная программа разработана для детей 12-17 лет. В группы для обучения специального отбора не производится. Принимаются все желающие.

Занятия построены с учетом возрастных психофизиологических особенностей детей, с учетом их индивидуальности, уровня подготовки и другим индивидуальным особенностям.

Оптимальное количество обучающихся в объединении для успешного освоения программы 10-15 человек.

Сроки реализации программы:

Дополнительная общеобразовательная общеразвивающая программа «Программирование: Python» рассчитана на 1 год обучения (72 часа)

Режим занятий:

2 академических часа в неделю. Время занятий: 1 ак. час – 40 минут, с перерывом на отдых – 10 минут.

Цель программы: развитие алгоритмического и логического мышления посредством языка программирования «Python».

Задачи:

– Обучающие:

- сформировать у обучающихся представление об основных элементах программирования;
- познакомить с синтаксисом языка программирования Python;
- сформировать у обучающихся навыки работы в интегрированной среде разработки на языке Python;
- способствовать приобретению навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python у обучающихся.

Развивающие:

- совершенствовать аналитические навыки;
- формировать навык алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование информации при решении задач;
- развивать умение планировать свои действия с учётом фактора времени;

– Воспитательные:

- воспитывать в обучающихся усидчивость, аккуратность, умение доводить начатое дело до конца;
- формировать коммуникативные навыки.

Формы подведения итогов реализации программы

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: промежуточную и итоговую аттестацию результатов обучения детей.

Промежуточная аттестация проводится в конце первого полугодия.

Итоговый контроль проводится в конце года с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения.

Аттестация обучающихся может проходить на итоговом занятии в форме презентации своего проекта.

Формы организации учебного занятия:

- вводное занятие – педагог знакомит обучающихся с техникой безопасности, особенностями организации деятельности и предлагаемым планом работы на текущий год;
- ознакомительное занятие – педагог знакомит обучающихся с новыми методами работы в зависимости от темы занятия;

- тематическое занятие - на котором детям предлагается работать над моделированием по определенной теме. Занятие содействует развитию творческого воображения обучающихся;
- занятие-проект – на таком занятии обучающиеся получают полную свободу в выборе направления работы, не ограниченного определенной тематикой. Обучающиеся, участвующие в работе по выполнению предложенного задания, рассказывают о выполненной работе, о ходе выполнения задания, о назначении выполненного проекта;
- конкурсное игровое занятие – строится в виде соревнования для повышения активности обучающихся и их коммуникации между собой;
- комбинированное занятие – проводится для решения нескольких учебных задач;
- итоговое занятие – служит подведению итогов работы за учебный год, может проходить в виде мини-выставок, просмотров творческих работ и презентаций.

Алгоритм учебного занятия:

1. Мотивация обучающихся.
2. Актуализация имеющихся знаний.
3. Теоретический блок нового материала.
4. Закрепление материала.
5. Перерыв.
6. Теоретический блок нового материала.
7. Закрепление материала.
8. Рефлексия.

Планируемые результаты

К концу реализации программы обучающиеся будут знать:

- синтаксис языка программирования Python;
- простые и сложные структуры данных, а также конструкции для работы с ними;
- некоторые модули (random, math, collections, itertools и др.).
- основные элементы программирования: ввод (вывод) информации в память (из памяти), данные, операции с данными, условное выполнение, циклы, подпрограммы
- алгоритмы обработки данных.

К концу реализации программы обучающиеся будут уметь:

- анализировать предложенный алгоритм, например определять, какие результаты возможны при заданном множестве исходных значений и при каких исходных значениях возможно получение указанных результатов;
- создавать, анализировать и реализовывать в виде программ базовые алгоритмы, связанные с анализом элементарных функций (в том числе приближенных вычислений), записью чисел в позиционной системе счисления, делимостью целых чисел; линейной обра-

боткой последовательностей и массивов чисел (в том числе алгоритмы сортировки), анализом строк, а также рекурсивные алгоритмы;

- создавать собственные алгоритмы для решения прикладных задач на основе изученных алгоритмов и методов;
- применять при решении задач структуры данных: списки, применять при составлении алгоритмов базовые операции со структурами данных;
- использовать основные понятия, конструкции и структуры данных последовательного программирования, а также правила записи этих конструкций и структур в выбранном для изучения языке программирования;
- использовать в программах данные различных типов; применять стандартные и собственные подпрограммы для обработки символьных строк; выполнять обработку данных, хранящихся в виде массивов различной размерности; выбирать тип цикла в зависимости от решаемой подзадачи; составлять циклы с использованием заранее определенного инварианта цикла; реализовывать решения подзадач в виде подпрограмм, связывать подпрограммы в единую программу; использовать модульный принцип построения программ; использовать библиотеки стандартных подпрограмм;
- выполнять отладку и тестирование программ в выбранной среде программирования; использовать при разработке программ стандартные библиотеки языка программирования и внешние библиотеки программ; создавать многокомпонентные программные продукты в среде программирования

К концу реализации программы обучающиеся будут иметь представление:

- о возможностях языка программирования Python;
- о структурном программировании.

Организационные моменты работы:

Нагрузка и расписание: 72 учебных часа, (2 час/нед.) Место проведения: кабинет №22

Учебный план
72 академических часа в год

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
1	Введение в язык программирования Python	22	8	14
1.1	Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ	2	1	1
1.2	Вывод данных на языке Python. Ввод и отладка программ в среде	2	1	1
1.3	Переменные	2	1	1
1.4	Выражения. Типы данных	2	1	1
1.5	Строки	2	1	1
1.6	Методы строк	2		2
1.7	Списки	4	2	2
1.8	Методы списков	2		2
1.9	Словари	2	1	1
1.10	Методы словарей	2		2
2	Основы языка Python	38	16	22
2.1	Условный оператор	2	1	1
2.2	Объединение условий	2	1	1
2.3	Использование цикла for	3	1	2
2.4	Использование цикла while	3	1	2
2.5	Решение задач на циклические алгоритмы	4	1	3
2.6	Применение функций	2	1	1
2.7	Использование встроенных функций	4	1	3
2.8	Применение модулей	2	1	1
2.9	Классы и объекты	2	1	1
2.10	События	2	1	1
2.11	Работа с файлами	4	1	3
2.12	Решение задач на трассировку программ	8	-	8
3	Внешние библиотеки	8	4	4
3.1	Модули. Подключение модулей	2	1	1
3.2	Установка дополнительных модулей	2	1	1
3.3	Библиотека NumPy	2	1	1
3.4	Использование ресурсов дополнительных модулей	2	1	1
4	Аттестация	4	-	4
	Итого:	72	28	44

Содержание

Раздел 1. Введение в язык программирования Python

Тема 1.1. Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ

Теория. Техника безопасности на занятии. Понятие «алгоритм», «исполнитель», «язык программирования», «программа», «интерпретатор». История языка программирования Python и его возможности. Виды окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей.

Практика. Сохранение и запуск python-программ в среде разработки IDLE.

Тема 1.2. Вывод данных на языке Python. Ввод и отладка программ в среде

Теория. Правила синтаксиса Python: правило начала, правило порядка, правило регистра. Понятие функции. Функция print(). Форматный вывод

Практика. Проект «Символьная графика». Создание определённого рисунка с помощью символов. Отработка функции print().

Тема 1.3. Переменные

Теория. Понятие «переменная». Правила именования переменных в языке Python. Оператор присваивания. Функция input().

Практика. Ввод переменных. Изменение типа переменной.

Тема 1.4. Выражения. Типы данных

Теория. Арифметические операции с помощью математических операторов +, -, *, /. Порядок выполнения операций. Понятие «выражение», «типы данных». Функции int() и str().

Практика. Проект «Сумматор». При написании данной программы отрабатываются математические операторы и функции int() и str().

Тема 1.5. Строки

Теория. Понятие «строка». Создание строк. Переменные внутри строк. Операции со строками

Практика. Решение задач на ввод и обработку строк.

Тема 1.6. Методы строк

Теория. Методы строк

Практика. Решение задач на методы строк. Проект «Письмо» - форматирование исходного текста по правилу.

Тема 1.7. Списки

Теория. Понятие «список». Создание списков. Добавление/удаление элементов в/из список/списка. Операции со списками.

Практика. Решение задач на отработку операций со списками.

Тема 1.8. Методы списков

Теория. Методы списков

Практика. Решение задач на использование методов.

Тема 1.9. Словари

Теория. Понятие «словарь». Создание словаря.

Практика. Решение задач на создание словарей, добавления элементов, обращения к элементам

Тема 1.10. Методы словарей

Теория. Методы словарей

Практика. Проект «Любимые вещи»: создание списка любимых развлечений и любимых лакомств.

Тема 1.7. Рисование с помощью «Черепашьей графики»

Раздел 2. Основы языка Python

Тема 2.1. Условный оператор

Теория. Понятие «условный оператор», «вложенные команды», «оператор сравнения». Конструкция if и её синтаксис. Операторы сравнения: <, >, >=, <=, !=, ==. Структура программы. Конструкция if-else. Команды if и elif. Практика. Решение задач на обработку условного оператора и операторов сравнения.

Тема 2.2. Объединение условий

Теория. Логические операторы: and, or, not. Порядок выполнения операций. Переменные без значения - None.

Практика. Проект «Калькулятор»: создание приложения по определенным условиям.

Тема 2.3. Использование цикла for

Теория. Понятие «цикл», «цикл со счётчиком». Конструкция for и её синтаксис.

Практика. Проект «Таблица умножения»: создание приложения по определенным условиям.

Тема 2.4. Использование цикла while

Теория. Понятие «цикл с предусловием». Конструкция while и её синтаксис. За цикливание и выход из цикла с помощью команды break.

Практика. Проект «Бомба взорвалась!». Написание программы по определенным условиям.

Тема 2.5. Решение задач на циклические алгоритмы

Теория. Виды циклов и их конструкции.

Практика. Решение задач на применение циклов for и while.

Тема 2.6. Применение функций

Теория. Понятие «функция», «параметр функции», «значение функции». Строение функции: имя, аргумент, тело. Создание и вызов функции. Переменные и область видимости.

Практика. Решение задач на обработку понятия «функция», её строение и синтаксис.

Тема 2.7. Использование встроенных функций

Теория. Функции: abs, bool, dir, eval, exec, float, int, len, max, min, range, sum. Практика. Решение задач на отработку функций.

Тема 2.8. Применение модулей

Теория. Понятие «модуль». Импортинг модуля в программу. Полезные модули: random, time, pickle.

Практика. Решение задач на применение модулей.

Тема 2.9. Классы и объекты

Теория. Понятие «объект». Концепция объектов. Понятие «класс». Использование классов в Python. Инициализация объектов. Практика.

Решение задач на отработку умения создавать классы и объекты.

Тема 2.10. События

Теория. Понятие «событие», «подписка на событие», «обработка события».

Практика. Проект «Поймай черепашку». Пишется приложение-игра, в котором создается холст и на нём прорисовывается черепашка, при нажатии на которую происходит ее перемещение в новое место на холсте.

Тема 2.11. Работа с файлами

Теория. Понятие «файл». Классификация файлов в зависимости от видов информации. Создание текстового файла. Открытие файла в Python. Запись в файл.

Практика. Проект «Блокнот»: создание простейшего текстового редактора в виде приложения.

Тема 2.12. Решение задач на трассировку задач

Практика. Чтение программы. Определение результата выполнения программы по входным параметрам. Определение входных параметров для получения определенного результата работы программы

Раздел 3. Внешние библиотеки

Тема 3.1. Модули. Подключение модулей

Теория. Внешние модули Python.

Практика. Подключение внешних модулей. Решение задач на подключение модулей и использования их подпрограмм

Тема 3.2. Установка дополнительных модулей

Теория. Установка дополнительных модулей.

Практика. Использование модуля requests

Тема 3.3. Библиотека NumPy

Теория. Библиотека NumPy, установка, состав, возможности.

Практика. Решение аналитических задач на использование библиотеки NumPy

Тема 3.4. Использование ресурсов дополнительных модулей

Теория. Библиотека Matplotlib, ScyPy, установка, состав, возможности

Практика. Решение задач на использование подпрограмм библиотек Matplotlib, ScyPy

Раздел 4. Аттестация

Практика. Практикум по решению задач курса

Условия реализации программы

Материально-техническое обеспечение программы

- столы, стулья (по росту и количеству детей);
- демонстрационная маркерная доска;
- интерактивный комплекс;
- ноутбуки (15 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО.

Формы аттестации (контроля)

Входящий контроль осуществляется при комплектовании группы в начале учебного года. Цель - определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися. Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

Итоговый контроль осуществляется в конце учебного года. Форма контроля: контрольный практикум.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

Учебно-методическое обеспечение

- Поляков К.Ю. Программирование. Python. C++.: учебное пособие в четырех частях/К.Ю. Поляков. – М.: БИНОМ. Лаборатория знаний, 2019
- Компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещённый на сайте авторского коллектива: <http://kpolyakov.spb.ru/school/probook.htm>
- Электронный задачник-практикум с возможностью автоматической проверки решений задач по программированию: <http://informatics.mccme.ru/>

- Курс «Алгоритмы: теория и практика. Методы» образовательной платформы Stepik.org
- Курс «Программирование на Python» образовательной платформы Stepik.org

Также предполагается использование дополнительных пособий по программированию.

-- Python 3 для начинающих <https://pythonworld.ru/samouchitel-python>