

Управление образования Администрация Кулебакского района
Муниципальное бюджетное образовательное учреждение
школа № 7

Принята на заседании педагогического совета
Протокол №1 от 30.08.2023

Утверждена
Приказом МБОУ школы №7
от 01.09.2023 №322



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

Возраст обучающихся – 15 – 17 лет,
Срок реализации: 1 год

Составитель: педагог
дополнительного образования
Кузнецова Татьяна Олеговна

Кулебаки
2023

Оглавление

№	Структура	Стр.
1	Пояснительная записка	3
2	Содержание программы	4
2.1	Цель, задачи, планируемые результаты	5
2.2	Календарный учебный график	8
2.3	Учебный план	9
3	Рабочие программы учебных модулей	9
3.1	Содержание учебного модуля (1 полугодие)	9
3.2	Тематическое планирование учебного модуля (1 полугодие)	11
3.3	Содержание учебного модуля (2 полугодие)	13
3.4	Тематическое планирование учебного модуля (2 полугодие)	14
4	Оценочные материалы	15
5	Методические материалы	15
6	Перечень основного оборудования	15
7	Литература, используемая при организации образовательного процесса	15

1.Пояснительная записка

1.1 Направленностьдополнительной общеобразовательной (общеразвивающей) программы.

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 обучающимися 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.

1.2 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 прикладных задач, позволит потом с легкостью выучить любой другой язык программирования.

1.3 Педагогическая целесообразность программы

Педагогическая целесообразность программы заключается не только в развитии технических способностей и возможностей средствами конструктивно-технологического подхода, гармонизации отношений ребенка и окружающего мира, но и в развитии созидательных способностей, устойчивого противостояния любым негативным социальным и социотехническим проявлениям. 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;

2. Содержание программы

2.1 Цели, задачи и планируемые результаты

Цель программы: формирование устойчивого интереса к занятиям в сфере программирования, развитие алгоритмического и логического мышления.

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;
 - формировать учебную мотивацию и мотивацию к творческому поиску; развивать аккуратность, внимание и самоконтроль;
 - развивать умение планировать свои действия с учётом фактора времени; □
- Воспитательные:
- воспитывать дисциплинированность, ответственность, самоорганизацию; ○
 - воспитывать трудолюбие, уважение к труду; ○ формировать чувство коллективизма и взаимопомощи;
 - воспитывать чувство патриотизма, гражданственности, гордости за достижения ○
 - отечественной науки и техники.

Режим работы

Год обучения	Всего часов			
	В год	В неделю	В день	Перерыв между занятиями
1	68 ч	2 ч	-	10 минут

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 обработкой 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.

Основные подходы - организация практической деятельности, самостоятельная работа обучающихся, индивидуальные консультации.

При организации занятий используется дифференцированный, индивидуальный подход к каждому ребенку. Работа на занятии может быть групповая, по подгруппам, в парах, индивидуально.

Педагогические технологии

В процессе обучения по Программе используются разнообразные педагогические технологии:

- технологии развивающего обучения, направленные на общее целостное развитие личности, на основе активно-деятельного способа обучения, учитывающие закономерности развития и особенности индивидуума;
- технологии личностно-ориентированного обучения, направленные на развитие индивидуальных познавательных способностей каждого ребенка, максимальное выявление, раскрытие и использование его опыта;
- технологии дифференцированного обучения, обеспечивающие обучение каждого обучающегося на уровне его возможностей и способностей; технологии сотрудничества, реализующие демократизм, равенство, партнерство в отношениях педагога и обучающегося;
- проектные технологии – достижение цели через детальную разработку проблемы, которая должна завершиться реальным, осязаемым практическим результатом, оформленным тем или иным образом;
- компьютерные технологии, формирующие умение работать с информацией, исследовательские умения, коммуникативные способности.

В практике выступают различные комбинации этих технологий, их элементов.

Формы организации учебного занятия:

При проведении занятий используются разные формы работы:

- беседа;
- практическая работа;
- хакатон

Формы подведения итогов:

1. индивидуальная устная/письменная проверка;
2. фронтальный опрос, беседа; межгрупповые соревнования;
3. проведение промежуточного и итогового тестирования;

Уровни освоения программы:

Итоговая оценка развития личностных качеств воспитанника производится по трём уровням:

1. «высокий»: положительные изменения личностного качества воспитанника в течение учебного года признаются как максимально возможные для него;
2. «средний»: изменения произошли, но воспитанник потенциально был способен к

4.большому;

5.«низкий»: изменения не замечены.

2.2 Календарный учебный график

Год/месяц/неделя		Год обучения/ Кол-во часов
Год обучения		1
сентябрь		
1 неделя	1.09- 5.09	2
2 неделя	6.09.-12.09	2
3 неделя	19.09-19.09	2
4 неделя	20.09-26.09	2
5 неделя	27.09-03.10	2
октябрь		
1 неделя	4.10-10.10	2
2 неделя	11.10-17.10	2
3 неделя	18.10-24.10	2
4 неделя	25.10-31.10	2
1 неделя	1.11-7.11	2
2 неделя	8.11-14.11	2
3 неделя	15.11-21.11	2
4 неделя	22.11-28.11	2
декабрь		
1 неделя	29.12-5.12	2
2 неделя	6.12-12.12	2
3 неделя	19.12-19.12	2
4 неделя	20.12-26.12	2
5 неделя	27.12-2.01	2
январь		
1 неделя	3.01-9.01	2
2 неделя	11.01-16.01	2
3 неделя	18.01.-23.01	2
4 неделя	25.01-30.01	2
февраль		
1 неделя	31.01-6.02	2
2 неделя	7.02-13.02	2
3 неделя	14.02-20.02	2
4 неделя	21.02-27.02	2
Март		
1 неделя	28.02-6.03	2
2 неделя	7.03-13.03	2
3 неделя	14.03-20.03	2
4 неделя	21.03-27.03	2
5 неделя	28.03-3.04	2
апрель		
1 неделя	4.04-10.04	2

2 неделя	11.04-17.04	2
3 неделя	18.04-24.04	2
4 неделя	25.04-1.05	2
май		
1 неделя	2.05-8.05	2
2 неделя	9.05-15.05	2
3 неделя	16.05-22.05	2
4 неделя	23.05-29.05	2
5 неделя	30.05-5.06	2
Всего:	38 недель	76

	Ведение занятий по расписанию
	Каникулярный период
	Промежуточная аттестация
	Выходные праздничные дни

2.3 Учебный план

№ п/п	Название разделов		
		Кол-во часов	Формы промежуточной аттестации
1	Учебный модуль (1 полугодие)	34	Выполнение проекта
2	<i>Промежуточная аттестация</i>		
3	Учебный модуль (2 полугодие)	34	Выполнение проекта
4	<i>Промежуточная аттестация</i>		

3. Рабочие программы учебных модулей

3.1. Содержание учебного модуля 1 полугодие

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 «Черепашей 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 отработку 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. Зацикливание 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.

3.2. Тематическое планирование учебного модуля (1 полугодие)

№ п/п	Название разделов, тем	Количество часов			Формы контроля
		Всего	Теория	Практика	
1	Введение в язык программирования Python	20	7	13	Выполнение практических задач
1.1	Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ	2	1	1	Выполнение практических задач
1.2	Вывод данных на языке Python. Ввод и отладка программ в среде	2	1	1	Выполнение практических задач

1.3	Переменные	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
1.4	Выражения. Типы данных	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
1.5	Строки	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
1.6	Методы строк	2		2	Выполнение лабораторных работ, практических задач, кейс-заданий
1.7	Списки	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
1.8	Методы списков	2		2	Выполнение лабораторных работ, практических задач, кейс-заданий
1.9	Словари	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
1.10	Методы словарей	2		2	Выполнение лабораторных работ, практических задач, кейс-заданий
2	Основы языка Python	13	5	8	
2.1	Условный оператор	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.2	Объединение условий	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.3	Использование цикла for	3	1	2	Выполнение лабораторных работ, практических задач, кейс-заданий
2.4	Использование цикла while	3	1	2	Выполнение лабораторных работ, практических задач, кейс-заданий
2.5	Решение задач на циклические алгоритмы	4	1	2	Выполнение лабораторных работ, практических задач, кейс-заданий
3	Промежуточная аттестация	1			
Итого		34			

3.3.Содержание учебного модуля 2 полугодие

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 отработку 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». Импорт 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

Проект «Разработка программного обеспечения для решения
практикоориентированной задачи»

3.4. Тематическое планирование учебного модуля (2 полугодие)

№ п/п	Название разделов, тем	Количество часов			Формы контроля
		Всего	Теория	Практика	
2	Основы языка Python	24	7	17	
2.5	Решение задач на циклические алгоритмы	2	1	2	Выполнение лабораторных работ, практических задач, кейс-заданий
2.6	Применение функций	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.7	Использование встроенных функций	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.8	Применение модулей	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.9	Классы и объекты	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.10	События	2	1	1	Выполнение лабораторных работ, практических задач, кейс-заданий
2.11	Работа с файлами	4	1	3	Выполнение лабораторных работ, практических задач, кейс-заданий
2.12	Решение задач на трассировку программ	9	-	9	Выполнение практических задач
3	Внешние библиотеки	8	4	4	

3.1	Модули. Подключение модулей	2	1	1	Выполнение практических задач
3.2	Установка дополнительных модулей	2	1	1	Выполнение практических задач
3.3	Библиотека NumPy	2	1	1	Выполнение практических задач
3.4	Использование ресурсов дополнительных модулей	2	1	1	Выполнение практических задач
4	Промежуточная аттестация	2	-	4	Выполнение проекта
	Итого:	34	11	23	

4. Оценочные материалы

Промежуточная аттестация по итогам освоения дополнительной общеобразовательной программы проводится по итогам 1 и 2 полугодий в форме защиты проекта.

Уровень освоения программы по итогам изучения учебных модулей оценивается по 3-м уровням:

Высокий уровень - с количеством баллов-10-8

Средний уровень-с количеством баллов за работу 6-7

Низкий уровень –с количеством баллов за работу менее 5

5. Методические материалы

В качестве методов обучения по программе используются наглядно-практический, исследовательские методы. На занятиях используются различные формы организации образовательного процесса.

Предполагается использование дополнительных пособий по программированию.

Курс «Программирование на Python» образовательной платформы Stepik.org

6. Перечень основного оборудования

Материально-техническое обеспечение программы

1. столы, стулья (по росту и количеству детей);

2. демонстрационная маркерная доска;

3. интерактивный комплекс;

4. ноутбуки (15 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО.

7. Литература, используемая при организации образовательного процесса

1. Поляков К.Ю. Программирование. Python. C++.: учебное пособие в четырех частях/К.Ю. Поляков. – М.: БИНОМ. Лаборатория знаний, 2019

2.Компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещённый на сайте авторского коллектива:

<http://kpolyakov.spb.ru/school/probook.htm>

3.Электронный задачник-практикум с возможностью автоматической проверки решений задач по программированию: <http://informatics.mccme.ru/>

4.Курс «Алгоритмы: теория и практика. Методы» образовательной платформы Stepik.org