

Управление образования муниципального образования
Красноармейский район

Муниципальное автономное общеобразовательное учреждение средняя
общеобразовательная школа № 10 станицы Новомышастовской

Принято на заседании
методического совета
от 31.08.2020 г.
Протокол № 1-

Утверждаю
Директор МАОУ СОШ № 10
Е.И.Вервыкишко
_01.09.2020 год



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«Программирование на языке PYTHON»**

Уровень образования: ознакомительная

Срок реализации программы: 1 год

Возрастная категория : 16-17 лет

Вид программы: модифицированная

Программа реализуется на бюджетной основе

ID программы на сайте «Навигатор»: 14065

Автор –составитель:
учитель информатики
Кузьмин Д.И.

Ст.Новомышастовская 2020 г.

СОДЕРЖАНИЕ

Раздел №1 «Комплекс основных характеристик программы»

- 1.1. Пояснительная записка.....
- 1.2. Содержание программы.....

Раздел №2 «Комплекс организационно-педагогических условий»

- 2.1. Календарный учебный график
- 2.2. Условия реализации программы...
- 2.3. Форма аттестации.....
- 2.4. Оценочные материалы.....
- 2.5. Методические материалы.....
- 2.6. Список литературы.....

«Комплекс основных характеристик программы»

1.1. Пояснительная записка

Дополнительная общеобразовательная (общеразвивающая) программа **«Программирование на языке PYTHON»** 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 интернет-ресурсы 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.

Возрастная категория обучающихся по программе от 16 до 17 лет.

Срок реализации программы составляет 1 год. Общее количество часов в год составляет 36 часов.

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 40 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.

1.2. Содержание программы.

Учебный (тематический) план

Учебно-тематический план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	Введение в образовательную программу, техника безопасности	2	2	-	Тестирование
2.	Основы языка Python. Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных	8	2	6	Тестирование
3.	Кейс 1. «Угадай число»	12	5	7	Демонстрация решений кейса

3.1	Введение в искусственный интеллект. Примеры на языке Python с искусственным интеллектом по угадыванию чисел, метод дихотомии. Управление искусственным интеллектом	8	3	5	
3.2	Подготовка к публичному выступлению для защиты результатов. Демонстрация отчёта в группе и защита результатов работы	4	2	2	
4	Кейс 4. Программирование автономных квадрокоптеров	2	9	13	Демонстрация решений кейса
4.1	Техника безопасности при полётах. Проведение полётов в ручном режиме	4	1	3	
4.2	Программирование взлёта и посадки беспилотного летательного аппарата	12	5	7	
4.3	Выполнение команд «разворот», «изменение высоты», «изменение позиции»	4	2	2	
4.4	Выполнение группового полёта вручную	6	0	6	
4.5	Выполнение позиционирования по меткам	6	1	5	
	Итого:	36	18	18	

Содержание учебного (тематического) плана

Содержание учебно-тематического плана

№	Темы занятий	Содержание занятий
1.	Вводное занятие. Введение в предмет, техника безопасности (2 ч)	Теория: введение в образовательную программу. Ознакомление обучающихся с программой, приёмами и формами работы. Вводный инструктаж по ТБ.

2.	<p>Основы языка Python. Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных (8 ч)</p>	<p>Теория: история языка Python, сфера применения языка, различие в версиях, особенности синтаксиса. Объявление и использование переменных в Python. Использование строк, массивов, кортежей и словарей в Python. Использование условий, циклов и ветвлений в Python.</p> <p>Практика: запуск интерпретатора. Различия интерпретатора и компилятора. Написание простейших демонстрационных программ. Минипрограммы внутри программы. Выражения в вызовах функций. Имена переменных. Упражнения по написанию программ с использованием переменных, условий и циклов. Генерация случайных чисел. Группировка циклов в блоки. Операции сравнения.</p>
3	<p>Кейс «Угадай число»</p>	
3.1	<p>Введение в искусственный интеллект. Примеры на языке Python с искусственным интеллектом по угадыванию чисел, метод дихотомии. Управление искусственным интеллектом (12 ч)</p>	<p>Теория: алгоритмы поиска числа в массиве. Варианты сортировок. Поиск дихотомией. Работа с переменными, работа с функциями.</p> <p>Практика: упражнения по поиску чисел в массиве. Упражнения на сортировку чисел. Алгоритмы поиска числа. Исследование скорости работы алгоритмов.</p>
3.2	<p>Подготовка к публичному выступлению для защиты результатов. Демонстрация отчёта в группе и защита результатов работы (4 ч)</p>	<p>Теория: создание удобной и понятной презентации.</p> <p>Практика: подготовка презентации для защиты. Подготовка речи для защиты.</p>
4	<p>Кейс «Программирование автономных квадрокоптеров»</p>	
4.1	<p>Техника безопасности при полётах. Проведение полётов в ручном режиме (4 ч)</p>	<p>Теория: знакомство с кейсом, представление поставленной проблемы, правила техники безопасности. Изучение конструкции квадрокоптеров.</p> <p>Практика: полёты на квадрокоптерах в ручном режиме.</p>
4.2	<p>Программирование взлёта и посадки беспилотного летательного аппарата (8 ч)</p>	<p>Теория: основы программирования квадрокоптеров на языке Python. Практика: тестирование написанного кода в режимах взлёта и посадки.</p>

4.3	Выполнение команд «разворот», «изменение высоты», «изменение позиции» (12 ч)	Теория: теоретические основы выполнения разворота, изменения высоты и позиции на квадрокоптерах. Практика: тестирование программного кода в режимах разворота, изменения высоты и позиции.
4.4	Выполнение группового полёта вручную (4 ч)	Практика: выполнение группового полёта на квадрокоптере в ручном режиме.
4.5	Выполнение позиционирования по меткам (14 ч)	Теория: основы позиционирования indoor и outdoor квадрокоптеров. Практика: тестирование режима позиционирования по ArUco - маркерам.

8.1. Защита индивидуального проекта.

Организационно-педагогические условия реализации программы.

Занятия проводятся в следующих формах: лекции, семинары, практические задания.

Дидактический материал, необходимый для проведения занятий:

- краткие конспекты материалов для лекций;
- распечатки заданий для практикумов;
- презентационные материалы для объяснения;
- карточки с индивидуальными заданиями.

Техническое оснащение занятий:

- компьютер для демонстрации презентаций;
- проектор;
- рабочие компьютеры учащихся для работы с доступом в Интернет;
- принтер для распечатки заданий.

Ожидаемые результаты и способы определения их результативности

По итогам реализации программы, учащиеся

будут знать:

- принципы программирования на языке Python;
- основы дизайнерского оформления созданных программ.

По итогам реализации программы, учащиеся

будут уметь:

- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.

Формы определения результативности обучения

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий.

Формы аттестации и оценочные материалы

В процессе реализации программы предусмотрены следующие виды контроля:

- *входной контроль проводится* с целью определения уровня знаний учащихся (Приложение № 1);
- *промежуточный контроль* проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;
- *итоговый контроль* – защита проекта.

Формы подведения итогов реализации программы

Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта.

РАЗДЕЛ №2 «Комплекс организационно-педагогических условий.

2.1. Календарный учебный график

Тематическое планирование.

№ п/п	Тема занятия	Кол-во часов	Содержание занятий
1.	Введение в образовательную программу, техника безопасности	1	Теория: введение в образовательную программу. Ознакомление обучающихся с программой, приёмами и формами работы. Вводный инструктаж по ТБ.
2.	Основы языка Python. Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных	1	Теория: история языка Python, сфера применения языка, различие в версиях, особенности синтаксиса. Объявление и использование переменных в Python. Использование строк, массивов, кортежей и словарей в Python.

3.	Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных	1	Использование условий, циклов и ветвлений в Python. Практика: запуск
4.	Основы языка Python.	1	
5.	Основы языка Python.	1	

			интерпретатора. Различия интерпретатора и компилятора. Написание простейших демонстрационных программ. Минипрограммы внутри программы. Выражения в вызовах функций. Имена переменных. Упражнения по написанию программ с использованием переменных, условий и циклов. Генерация случайных чисел. Группировка циклов в блоки. Операции сравнения.
6.	Кейс 1. «Угадай число»	1	Теория: алгоритмы поиска числа в массиве. Варианты сортировок. Поиск дихотомией. Работа с переменными, работа с функциями. Практика: упражнения по поиску чисел в массиве. Упражнения на сортировку чисел. Алгоритмы поиска числа. Исследование скорости работы алгоритмов.
7.	Кейс 1. «Угадай число»	1	
8.	Кейс 1. «Угадай число»	1	
9.	Кейс 1. «Угадай число»	1	
10.	Кейс 1. «Угадай число»	1	
11.	Кейс 1. «Угадай число»	1	
12.	Кейс 1. «Угадай число»	1	
13.	Кейс 1. «Угадай число»	1	

14.	Кейс 4. «Программирование автономных квадрокоптеров»	1	Теория: знакомство с кейсом, представление поставленной проблемы, правила техники безопасности. Изучение конструкции квадрокоптеров. Практика: полёты на квадрокоптерах в ручном режиме.
15.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
16.	Кейс 4. «Программирование автономных квадрокоптеров»	1	Теория: основы программирования квадрокоптеров на языке Python. Практика: тестирование написанного кода в режимах взлёта и посадки.
17.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
18.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
19.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
20.	Кейс 4. «Программирование автономных квадрокоптеров»	1	

21.	Кейс 4. «Программирование автономных квадрокоптеров»	1	Теория: теоретические основы выполнения разворота, изменения высоты и позиции на квадрокоптерах. Практика: тестирование программного кода в режимах разворота, изменения высоты и позиции.
22.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
23.	Кейс 4. «Программирование автономных квадрокоптеров»	1	

24.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
25.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
26.	Кейс 4. «Программирование автономных квадрокоптеров»	1	Практика: выполнение группового полёта на квадрокоптере в ручном режиме.
27	Кейс 4. «Программирование автономных квадрокоптеров»	1	
28..	Кейс 4. «Программирование автономных квадрокоптеров»	1	Теория: основы позиционирования indoor и outdoor квадрокоптеров. Практика: тестирование режима позиционирования по ArUco - маркерам.
29.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
30.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
	автономных квадрокоптеров»		
31.	Кейс 4. «Программирование автономных квадрокоптеров»	1	
32.	Кейс 4. «Программирование автономных квадрокоптеров»	1	

33.	Кейс 4. «Программирование автономных квадрокоптеров»	1	Практика: тестирование режима позиционирования по ArUco - маркерам
34 -36	Кейс 4. «Программирование автономных квадрокоптеров»	3	
Итого		36	

2.2. Условия реализации программы

Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 Гб DDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JatBrainsPyCharm;
- FoxitReader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

2.3. 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.

Промежуточный контроль осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

Итоговый контроль осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

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.

2.4. Оценочные материалы

Входящий контроль: *Тестирование (Приложение 1)*

Промежуточный контроль: *Тестирование, решение задач (Приложение 2)*

Итоговый контроль: *учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.*

2.5. Методическое обеспечение

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2020 года.

Методы обучения:

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.

3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.

4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Технологии:

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

2.6. Список литературы

Нормативно-правовые акты и документы

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ
2. Приказ Министерства образования и науки РФ от 29.08.2013 №1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Письмо Министерства образования Российской Федерации от 18 июня 2003 №28-02-484/16 «Требования к содержанию и оформлению образовательных программ дополнительного образования детей»,
4. Постановление Главного государственного санитарного врача Российской Федерации от 4.07.2014 г. «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Стратегия инновационного развития Российской Федерации на период до 2020 года. Распоряжение правительства Российской Федерации от 08.12.2011 № 2227-р.
6. Концепция развития дополнительного образования детей. Распоряжение Правительства Российской Федерации от 24 апреля 2015 года № 729-р.
7. «Стратегия развития воспитания в Российской Федерации на период до 2025 года». Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.

4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

Литература для родителей и учащихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.

2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.

3. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.

4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.