

УПРАВЛЕНИЕ ОБРАЗОВАНИЕМ АДМИНИСТРАЦИИ  
МУНИЦИПАЛЬНОГО ОБРАЗОВАНИЯ ПАВЛОВСКИЙ РАЙОН  
МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 6 ИМЕНИ  
ФЁДОРА ИВАНОВИЧА ЯРОВОГО СТАНИЦЫ НОВОЛЕУШКОВСКОЙ

УТВЕРЖДЕНО

решением педагогического совета  
от 30.08. 2023 года протокол № 1  
Председатель \_\_\_\_\_ Овдиенко И.А.  
подпись руководителя ОУ Ф.И.О.

ДОПОЛНИТЕЛЬНАЯ  
ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ

**кружка «Информатика»**

Уровень программы ознакомительный.

Срок реализации программы 1 год (34 часа).

Возрастная категория: от 14 до 15 лет.

Вид программы: модифицированная

Автор-составитель: Бахал Алеся Михайловна, педагог дополнительного образования.

Ст. Новолеушковская, 2023

## Паспорт программы

	Информатика	
	Возраст учащихся	14-15 лет
2	Срок обучения	1 год
3	Количество часов(общее)	1
4	Количество часов в год	34
5	Ф.И.О. педагога	Бахал Алеся Михайловна
6	Уровень программы	Ознакомительный
7	Продолжительность 1 -го занятия (по САНПИНу)	45 минут
8	Количество часов в день	1 час
9	Периодичность занятий в неделю	1 раз

## Содержание

1 .	Раздел 1 программы «Комплекс основных характеристик образования.	
1.1	Пояснительная записка программы.	
1.2	Цели и задачи.	
1 .3	Содержание программы.	
1.4	Планируемые результаты.	
2.	Раздел 2 программы «Комплекс организационно-педагогических условий»	
2.1	Календарный учебный график.	
2.2.	Условия реализации программы.	
2.3.	Формы аттестации.	
2.4.	Оценочные материалы.	
2.5.	Методические материалы.	
2.6.	Список литературы.	

## **Раздел № 1 программы** **«Комплекс основных характеристик: объем, содержание,** **планируемые результаты»**

### *1.1 Пояснительная записка программы*

Программа внеурочной деятельности «Основы 3D-моделирования» разработана на основе:

- Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ;

- Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р);

- СанПин к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41);

- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства образования и науки РФ от 29.08.2013 № 1008);

- Стратегия развития воспитания в Российской Федерации на период до 2025 года;

- Дополнительная общеразвивающая программа «Python для начинающих» Возраст учащихся: 12—18 лет Авторы курса: Белых Роман, Гув Тимур, Скобина Ольга. Заказчик: Благотворительный фонд развития образования “Айкью Опшн” (“Возможность Интеллекта”);

- Дополнительная общеразвивающая программа «Космическая вёрстка» Авторы курса: Нуриахметова Арина, Плоткина Мария, Лихобабин Глеб, Турапина Катерина. Заказчик: Благотворительный фонд развития образования “Айкью Опшн” (“Возможность Интеллекта”)

Программа данного курса посвящена обучению школьников различным аспектам программирования на современном языке Python и языку разметки HTML. Занятия курса направлены на развитие мышления, логики, творческого потенциала учеников. Программа ориентирована на использование получаемых знаний для разработки реальных проектов. Курс содержит большое количество творческих заданий (именуемых Кейсами).

### **Цель и задачи обучения**

#### *1.2 Цели и задачи программы*

Программа по предмету «Информатика» предназначена для изучения курса информатики учащимися основной школы.

Целью изучения предмета «Информатика» является:

- формирование у обучающихся навыков программирования на языке Python и самонаправленного обучения;

- формирование у обучающихся ИКТ компетенции применения знаний языков HTML и CSS для проектной деятельности по созданию сайтов.

Для достижения поставленных целей необходимо решение следующих задач:

- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
- формирование информационной и алгоритмической культуры;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей;
- овладение важнейшими общеучебными умениями и универсальными учебными действиями (формулировать цели деятельности, планировать ее, находить и обрабатывать необходимую информацию из различных источников, включая Интернет и др.).

### *1.3. Содержание программы*

#### **Основы языка программирования Python (8 часов)**

Ввод-вывод данных. Типы данных. Работа со строками. Списки. Условная инструкция. Циклы for и while. Функции. Модули. Работа с текстовыми файлами.

#### **Проектная деятельность (4 часа)**

Разработка несложных консольных приложений.

#### **Основы языка разметки HTML (18 часов)**

Принципы работы сети Интернет; знакомство с языками HTML и CSS и их структурой; изучение инструментов разработки веб-браузера; регистрация на платформе «HTML Academy» и выполнение практической работы; знакомство со стилями CSS и применение их на практике; установка и знакомство с программой Gimp; изучение основных инструментов программы; знакомство с процедурой размещения сайта в Интернете; знакомство с крупнейшим веб-ресурсом для хостинга GitHub и публикация проекта на этом ресурсе.

#### **Работа над индивидуальным сайтом (4 часа)**

Создание собственного сайта.

### Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Основы языка программирования Python.	8	2	6	Текущий
2.	Проектная деятельность	4		4	Текущий
3.	Основы языка разметки HTML	10	5	5	Текущий

4.	Процедура размещения готовых web-страниц в сети Интернет	8	2	6	Текущий
5.	Работа над индивидуальным сайтом	4		4	Текущий
	Итого:	34	9	25	

### Содержание учебного плана.

#### **Модуль 1. Основы языка Python.**

##### **Урок 1. Вводное занятие.**

Техника безопасности. Знакомство с программой курса. Краткая история языка Python, кто создал и почему так назвали. Что можно сделать, зная язык программирования Python? Где в принципе программисты могут писать программы и как они их потом запускают? Отличие компилируемых и интерпретируемых языков программирования.

Демонстрация установки интерпретатора Python. Используем Python как калькулятор: интерактивный режим работы с интерпретатором Python, вычисляем сумму чисел, вычисляем 2 в степени 100. Запуск встроенной среды разработки IDLE, работа в командной строке Shell – аналог интерактивного режима. Тест по технике безопасности и простым фактам о языке Python.

##### **Урок 2. Основы языка Python. Ввод-вывод данных, числа и строки, операции с числами и строками.**

Встроенная среда разработки IDLE. Знакомство с функциями print() и input(). Особенность ввода данных на языке Python – данные считываются в виде строки. Первая программа: считываем число и выводим его же. Простые операции со строками, конкатенация строк. Создание простой программы, спрашивающей имя пользователя и затем приветствующей его. Создание простой программы, спрашивающей у пользователя имя, количество лет, а затем выводящей имя столько раз, сколько пользователю лет.

##### **Урок 3. Основы языка Python. Операции с числами и строками.**

Типы данных int и str. Преобразование типов с помощью операторов int() и str(). Простые вычисления в Python, например, “Чему равно str(2 + 3) \* int(‘2’ + ‘3’)? Постарайтесь дать ответ, не используя интерпретатора Python”.

Особенности функций input() и print(). Изменение поведения функции print() с помощью параметров sep и end. Символ перевода строки ‘\n’. Тест на понимание работы функций print, input, str, int.

Простые программы на взаимодействие с системой. Два возможных примера описаны ниже.

##### **Пример 1. Поле Чудес**

Программа приветствует Буратино и спрашивает, сколько у него монет, после ответа пользователя программа сообщает Буратино, сколько денег у него будет завтра.

```
print(‘Привет, Буратино!’)
```

```
print("Зарой все свои деньги здесь.")
n = input("Сколько у тебя монет?")
n = int(n)
print("Завтра у тебя будет", n * 100, "монет")
print("Приходи завтра!")
```

### **Пример 2. Инопланетянин**

Программа приветствует пользователя, узнает его возраст и печатает соответствующее количество сердечек.

```
print("Привет, землянин!")
n = input("Сколько тебе лет?")
n = int(n)
print("Шлю тебе", n, "сердечек")
print("\u2764 * n)
```

### **Урок 4. Основы языка Python. Условная инструкция в Python.**

Условная инструкция if-else в Python. Блок-схема ветвления. Неполное и полное ветвление. Отступы в Python – почему это важно? Операторы сравнения ==, !=, <, <=, >, >=, <, >.

Простые программы на использование условной инструкции. Несколько возможных примеров описаны ниже. Предложите школьникам придумать самим подобные программы и реализовать их.

#### **Программа 1.**

```
print("На улице темно? да/нет")
ans = input()
if ans == 'да':
    print("спокойной ночи!")
```

#### **Программа 2.**

```
ans = input("У тебя есть щупальца? да/нет")
if ans == 'да':
    print("Привет, осьминог!")
else:
    print("Привет, человек!")
```

### **Урок 5. Основы языка Python. Условная инструкция в Python, elif, логические операции.**

Как быть, если одновременно нужно проверить истинность нескольких условий? Или то, что верно хотя бы одно условие из нескольких? На помощь приходят логические операции or и and. Простые программы, например,

#### **Программа 1. Пройти на аттракцион**

```
rost = input("Каков твой рост в сантиметрах?\n")
vozrast = input("Сколько тебе лет?\n")
if int(rost) > 120 and int(vozrast) >= 3:
    print("Проходи!")
else:
    print("Подрасти еще немного!")
```

Если нужно больше двух ветвлений, стоит использовать оператор elif. Показать несколько простых примеров с использованием elif и без использования elif. Создание простых программ, где удобно использовать elif, например,

## Программа 2. Погода

```
weather = input('Какая сегодня погода? дождь/снег/солнце')
if weather == 'дождь':
    print('Захватите зонтик!')
elif weather == 'снег':
    print('Не забудьте варежки!') else:
    print('Ура! Берем солнечные очки!')
```

Что такое модули и как их подключать? Модуль random и некоторые его функции. Примеры использования модуля random.

## Программа 3. Чье число больше?

```
import random
n = input('Загадай число от 1 до 5, и я тоже загадаю\n')
m = random.randrange(1, 10)
if m > n:
    print('Мое число', m, 'больше твоего, ура!')
else:
    print('Я загадал', m, 'и проиграл...')
```

**Вопрос к школьникам:** модернизируйте программу так, чтобы рассматривался и случай равенства чисел.

## Урок 6. Цикл for в Python.

Использование цикла для перебора объектов (конструкция for elem in object, где object – строка, кортеж, иной итерируемый объект). Использование цикла для выполнения заданного количества операций: три способа записи for i in range(n); for i in range(a, b); for i in range(a, b, d)). Простые программы, например:

- Как напечатать слово привет столько раз, какое число введет пользователь (каждый “привет” должен быть в отдельной строке)?
- Напечатайте числа от 1 до n (n определяется пользователем) в прямом и в обратном порядке, только четные, только кратные трем, и т.п.
- Пользователь задает число n, программа выводит n строк, в первой строке одна звездочка, во второй две и т.д.

## Урок 7. Цикл while

Цикл с предусловием. Блок-схема. Бесконечный цикл. Оператор break для выхода из цикла. Переменная-счетчик для подсчета количества операций в цикле. Простые программы, например, такие, как приведены ниже:

Программа 1. Сладкоежка

```
candies = 0
ans = 'да'
while ans == 'да':
    candies = candies + 1
    print('Съедено конфет:', candies)
    ans = input('Хочешь еще конфетку? да/нет\n')
```

## Программа 2. Бесконечный цикл:

```
while True:
    print('я буду работать вечно!')
```

Программа 3. Надоедливая программа:



```

while True:
    ans = input('я тебе не надоела? да/нет\n')
    if ans == 'да':
        print('Как некрасиво! Ухожу...')
        break

```

#### **Программа 4. Угадай, как меня зовут?**

```

print('Угадай, как меня зовут!')
ans = input()
while ans != 'python':
    print('Не угадал! Попробуй еще раз')
    ans = input()
print('Правильно, я python!')

```

#### **Урок 8. Строки**

Нумерация символов строк. Отрицательная нумерация. Функция длины строки. Перебор символов строки с помощью цикла for (два способа: for smb in str и по индексу). Срезы строк. Простые программы: напечатать заданную пользователем строку задом наперед; напечатать все символы заданной пользователем строки по одному в строке; изменить заданную пользователем строку так, чтобы символы были разделены знаком звездочка, и т.п.

#### **Урок 9. Кейс 1. Создание программы-теста из нескольких вопросов.**

Создание тематического теста. Программа ведет диалог с пользователем, задавая ему 1-3 вопроса, каждый по одной и той же схеме (задается вопрос, далее дается три попытки на ответ, если одна из попыток заканчивается верным ответом, выводится строка "Ответ верный!", если ни в одной попытке не было дано верного ответа, просто показывается верный ответ.) Программа должна засчитывать верный ответ без учета регистра, для реализации этого удобно использовать встроенную функцию lower или встроенную функцию upper. Пример подобной программы, задающей пользователю вопрос про футбол, приведен ниже.

```

print('Тест про футбол')
ans = input('Вопрос 1. В какой стране проходил последний чемпионат
мира по футболу\n')
checker = True
attempt = 0
while checker and attempt < 3:
    if ans.lower() == «россия» or ans.lower() == «в россия»:
        print('Ответ верный!')
        checker = False
    else:
        if attempt < 2:
            ans = input('Попробуй еще раз!\n')
            attempt = attempt + 1
        if attempt == 3:
            print('Попытки закончились... Верный ответ: Россия или В России')

```

## **Урок 10. Завершение работы над программой-тестом.**

Добавление в тест еще одного-двух вопросов. Добавление счетчика очков (за каждый верный ответ пользователю начисляется определенное количество очков, после прохождения теста это количество очков выводится на экран). Обсуждение со школьниками того, как использование функций могло бы помочь сделать код короче (все вопросы теста однотипны), реализация при достаточном уровне подготовленности школьников.

## **Урок 11. Создание простых приложений.**

Приложение “Радуга”: простое приложение, состоящее из кнопок, окрашенных в основные цвета радуги и текстового поля. При нажатии на кнопку, в текстовом поле появляется название цвета нажатой кнопки.

## **Урок 12. Создание простых приложений.**

Упаковщики виджетов: `pack()`, `grid()`, `place()`. Создание простого калькулятора из нескольких кнопок и текстового поля для вывода результата вычислений.

## **Модуль 2. Основы языка разметки HTML**

### **Урок № 13. Готовимся к полёту.**

Знакомство с принципами работы сети Интернет. Знакомство с языками HTML и CSS и их структурой. Изучение инструментов разработки веб-браузера. Регистрация на платформе «HTML Academy» и выполнение практической работы.

### **Урок № 14. Первый космический код. Работа с HTML**

Установка программы Brackets. Попытки написать первый код в HTML. Практическая работа с кодом и создание Дневника Белки Стрелки и практические задания в «HTML Academy»

### **Урок № 15. Знакомство с CSS. Украшаем HTML-код**

Стилизация Дневника Белки Стрелки со стилями CSS. Знакомство с контейнерами `div`. Практические упражнения на стилизацию.

### **Урок № 16. Космическое пространство. Флексбоксы**

Знакомство с флексбоксами и их применение на практике. Интерактивные упражнения в классе и на платформе. Продолжение написания кода, применение флексбоксов. Упражнения на закрепление флексбоксов.

### **Урок № 17. Космический макет в графическом редакторе**

Изучение основных инструментов программы. Работа с макетом и подготовка этого макета для верстки сайта.

### **Урок № 18. Конструируем сайт. Готовим разметку.**

Повторение изученных ранее тегов, знакомство с новыми базовыми тегами. Верстка сайта по готовому макету.

### **Урок № 19. Конструируем сайт. Блоки и сетка**

Знакомство с понятиями “поток” и “блочные модели”. Изучение размеров объектов на странице и величин отступов. Создание сетки на Flexbox. Выполнение практической работы с изменением параметров объектов

### **Урок № 20. Готовимся к полёту. Создаём сетку проекта**

Верстка сетки с макета. Формирование стиля строчных и блочных элементов. Перенос свойства из макета GIMP в код. Продолжение верстки своего сайта с макетом и его стилизация.

### **Урок № 21. Готовимся к полёту. Стилизация элементов**

Знакомство с псевдоклассами. Знакомство с добавлением форм и таблиц на страницу и их стилизация. Завершение вёрстки страницы по стилю и наполнению.

### **Урок № 22. Последние приготовления перед запуском**

Правки в соответствии с подготовленным чек-листом.. Подготовка проекта к публикации. Добавление интерактивных элементов на страницу. Проверка страницы на соответствие и ошибки

### **Урок № 23. 3-2-1... Поехали! Запуск сайта в интернете**

Знакомство с процедурой размещения сайта в интернете. Знакомство с крупнейшим веб-ресурсом для хостинга GitHub. Публикация проекта на GitHub

### **Урок № 24. Презентация космической вёрстки**

Повторение всего, что изучали в рамках курса.. Презентация своих работ. Проведение небольшого турнира по скоростной верстке. Последние изменения макета

### **Урок № 25. Web-профессии и где могут пригодиться знания, полученные на курсе**

Популярные профессии 21 века, и какими навыками стоит обладать. Просмотр видео. Обсуждение области применения полученных на курсе знаний. Выполнение упражнений по ТЗ от заказчика

### **Урок № 26. 12 правил UI\UX. Создание лучшего макета для своего сайта**

Чек-лист идеального макета. Обзоры худших и лучших работ веб-верстальщиков. Создание собственного макета по правилам хорошего оформления

### **Урок № 27. Верстаем сайт по собственному макету**

Вспомнить основы хорошей вёрстки. Основные требования и правила. Кейс. Вёрстка сайта

### **Урок № 28. Продолжение вёрстки и встраивание медиа и анимаций.**

Виды и форматы медиа, которые можно без проблем встраивать на сайт. Добавление медиа на сайт

### **Урок № 29. Как сделать красивую презентацию для своего проекта. Доработка и оптимизация.**

Основные правила хорошей презентации. Как грамотно представить свой проект, не используя при этом сотню слайдов и не загромождая их информацией. Кейс. Доработка сайта

### **Урок № 30. Защита проекта.**

Защита проекта

### **Урок № 31. Работа над индивидуальным сайтом**

Создание структуры для собственного сайта.

### **Урок № 32. Графический дизайн.**

Отработка графического дизайна сайта.

### **Урок № 33. Верстка сайта.**

Информационное наполнение сайта.

### **Урок № 34. Демонстрация результата**

Представление своего проекта.

#### *1.4. Планируемые результаты.*

**Важнейшими умениями/знаниями являются следующие:**

- умение пользоваться персональным компьютером и его периферийным оборудованием;
- умение следовать требованиям техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий;
- умение осуществлять взаимодействие посредством электронной почты, чата, форума;
- умение искать информацию с применением правил поиска (построения запросов), в компьютерных сетях, некомпьютерных источниках информации (справочниках и словарях, каталогах, библиотеках) при выполнении заданий и проектов по различным учебным дисциплинам;
- знакомство с основными конструкциями языка Python (условная инструкция, циклы, функции, списки, строки) на практических примерах;
- формирование умений разрабатывать несложные консольные приложения и приложения с графическим интерфейсом;
- знать и уметь применять HTML-теги, атрибуты и способы структурирования содержимого web-страниц для создания форматированных документов;
- знать и уметь применять основы CSS: значения, списки, цвета, шрифты и другие метрики форматирования;
- владеть навыками проверки и отладки кода web-документов;
- владеть навыками быстрого и качественного форматирования сложных web-документов;
- умение выбирать способ представления своего проекта с использованием соответствующих программных средств.

**Личностными результатами, формируемыми при изучении предмета информатика, являются:**

- формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики,

учитывающего социальное, культурное, языковое, духовное многообразие современного мира;

- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.

#### **Метапредметные результаты изучения предмета «Информатика»:**

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;

- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;

- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

#### **Предметные результаты изучения предмета «Информатика»:**

- формирование представления об основных изучаемых понятиях курса;

- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;

- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для решения конкретной задачи;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей, с использованием соответствующих программных средств обработки данных;
- знакомство с базовыми конструкциями языка Python; формирование умения придумывать алгоритмы и их реализовывать на языке Python;
- знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- знать и уметь применять HTML-теги, атрибуты и способы структурирования содержимого web-страниц для создания форматированных документов;
- знать и уметь применять основы CSS: значения, списки, цвета, шрифты и другие метрики форматирования;
- владеть навыками проверки и отладки кода web-документов;
- владеть навыками быстрого и качественного форматирования сложных web-документов;
- использование готовых прикладных компьютерных программ по выбранной специализации;
- развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

## Раздел 2 программы «Комплекс организационно-педагогических условий»

### 2.1. Календарный учебный график

№ п/п	Дата по плану	Дата по факту	Тема занятия	Кол - во часов	Время проведения занятий	Форма занятия	Место проведения	Форма контроля
Основы языка программирования Python - 8ч								
1.			Вводное занятие.	1		Ознакомление	Каб.№ 10	Текущий
2.			Основы языка Python. Ввод-вывод данных, числа и строки, операции с числами и строками.	1		Комбинированное	Каб.№ 10	Текущий
3.			Основы языка Python. Операции с числами и строками.	1		Комбинированное	Каб.№ 10	Текущий
4.			Основы языка Python. Условная инструкция в Python.	1		Комбинированное	Каб.№ 10	Текущий
5.			Основы языка Python. Условная инструкция в Python, elif, логические операции.	1		Комбинированное	Каб.№ 10	Текущий
6.			Цикл for в Python.	1		Комбинированное	Каб.№ 10	Текущий
7.			Цикл while	1		Комбинированное	Каб.№ 10	Текущий
8.			Строки	1		Комбинированное	Каб.№ 10	Текущий
Проектная деятельность – 4ч.								
9.			Кейс 1. Создание программы-теста из нескольких вопросов.	1		Комбинированное	Каб.№ 10	Текущий
10.			Завершение работы над программой-тестом.	1		Ознакомительный	Каб.№ 10	Текущий
11.			Создание простых приложений.	1		Комбинированное	Каб.№ 10	Текущий
12.			Создание простых приложений.	1		Комбинированное	Каб.№ 10	Текущий
Основы языка разметки HTML – 10ч.								
13.			Готовимся к полёту.	1		Комбинированное	Каб.№ 10	Текущий
14.			Первый космический код. Работа с HTML	1		Комбинированное	Каб.№ 10	Текущий
15.			Знакомство с CSS. Украшаем HTML-код	1		Комбинированное	Каб.№ 10	Текущий

№ п/п	Дата по плану	Дата по факту	Тема занятия	Кол - во часов	Время проведения занятий	Форма занятия	Место проведения	Форма контроля
16.			Космическое пространство. Флексбоксы	1		Комбинированное	Каб.№ 10	Текущий
17.			Космический макет в графическом редакторе	1		Комбинированное	Каб.№ 10	Текущий
18.			Конструируем сайт. Готовим разметку.	1		Комбинированное	Каб.№ 10	Текущий
19.			Конструируем сайт. Блоки и сетка	1		Комбинированное	Каб.№ 10	Текущий
20.			Готовимся к полёту. Создаём сетку проекта	1		Комбинированное	Каб.№ 10	Текущий
21.			Готовимся к полёту. Стилизация элементов	1		Комбинированное	Каб.№ 10	Текущий
22.			Последние приготовления перед запуском	1		Комбинированное	Каб.№ 10	Текущий
Процедура размещения готовых web-страниц в сети Интернет – 8ч.								
23.			3-2-1... Поехали! Запуск сайта в интернете	1		Комбинированное	Каб.№ 10	Текущий
24.			Презентация космической вёрстки	1		Комбинированное	Каб.№ 10	Текущий
25.			Web-профессии и где могут пригодятся знания, полученные на курсе	1		Комбинированное	Каб.№ 10	Текущий
26.			12 правил UI\UX. Создание лучшего макета для своего сайта	1		Комбинированное	Каб.№ 10	Текущий
27.			Верстаем сайт по собственному макету	1		Комбинированное	Каб.№ 10	Текущий
28.			Продолжение вёрстки и встраивание медиа и анимаций.	1		Комбинированное	Каб.№ 10	Текущий
29.			Как сделать красивую презентацию для своего проекта. Доработка и оптимизация.	1		Комбинированное	Каб.№ 10	Текущий
30.			Защита проекта.	1		Комбинированное	Каб.№ 10	Текущий
Работа над индивидуальным сайтом – 4 ч								
31.			Работа над индивидуальным сайтом	1		Комбинированное	Каб.№ 10	Текущий
32.			Графический дизайн	1		Комбинированное	Каб.№ 10	Текущий
33.			Верстка сайта	1		Комбинированное	Каб.№ 10	Текущий



<b>№ п/п</b>	<b>Дата по плану</b>	<b>Дата по факту</b>	<b>Тема занятия</b>	<b>Кол - во часов</b>	<b>Время проведения занятий</b>	<b>Форма занятия</b>	<b>Место проведения</b>	<b>Форма контроля</b>
34.			Демонстрация результата	1		Комбинированное	Каб.№ 10	Текущий

## 2.2. Условия реализации программы.

Материально — техническое оснащение занятий. Для занятий с детьми требуется просторное светлое помещение, отвечающее санитарно-гигиеническим нормам. Помещение должно быть сухое, с естественным доступом воздуха, легко проветриваемым, с достаточным дневным и вечерним освещением. Вечернее освещение лучше всего обеспечивает люминесцентными лампами, создающими бестеневое освещение, близкое к естественному.

### Оборудование кабинета

Для занятий в помещении имеются столы, стулья, проектор и персональный компьютер (для показа презентаций).

Красивое оформление учебного помещения, чистота и порядок в нем, правильно организованные рабочие места имеют большое воспитательное значение. Все это дисциплинирует ребят, способствует повышению культуры их труда и творческой активности.

### Оборудования необходимые для занятия

Оборудование, поступившее в рамках реализации федерального проекта «Современная школа» национального проекта «Образование» в МБОУ СОШ №6 им. Ф.И. Ярового ст. Новолеушковской в структурное подразделение Центр образования цифрового и гуманитарного профилей «Точка роста».

№ п/п	Наименование оборудования	Краткие примерные технические характеристики	Ед. изм.	Количество
1	2	3	4	5
1.1	Аддитивное оборудование			
1.1.1	3D-принтер	Тип принтера: FDM, FFF. Материал (основной): PLA. Количество печатающих головок: 1. Рабочий стол: с подогревом. Рабочая область (XYZ): от 180x180x180 мм. Максимальная скорость печати: не менее 150 мм/сек. Минимальная толщина слоя: не более 20 мкм. Закрытый корпус: наличие. Охлаждение зоны печати: наличие.	шт.	1
1.1.2	Пластик для 3D-принтера	Материал: PLA, совместимость с п. 1.1.1	шт.	10
1.2	Компьютерное оборудование			
1.2.1	МФУ (принтер, сканер, копир)	Тип устройства: МФУ. Цветность: черно-белый. Формат бумаги: не менее А4. Технология печати: лазерная. Разрешение печати: не менее 1200x1200 точек. Автоподатчик бумаги: наличие	шт.	1
1.2.2	Ноутбук мобильного класса	Форм-фактор: трансформер. Жесткая клавиатура: требуется. Наличие русской раскладки клавиатуры: требуется. Сенсорный экран: требуется. Угол поворота сенсорного экрана (в случае неотключаемой клавиатуры): 360 градусов. Диагональ сенсорного экрана: не менее 11 дюймов. Производительность процессора (по тесту PassMark - CPU BenchMark <a href="http://www.cpubenchmark.net">http://www.cpubenchmark.net</a> ): не менее 2100 единиц. Объем оперативной памяти: не менее 4 Гб. Объем накопителя SSD/eMMC: не менее 128 Гб. Время автономной работы от батареи: не менее 7 часов. Вес ноутбука: не более 1,45 кг. Стилус в комплекте поставки: требуется. Корпус ноутбука должен быть специально подготовлен для безопасного использования в учебном процессе (иметь защитное стекло повышенной прочности,	шт.	10

		выдерживать падение с высоты не менее 700 мм, сохранять работоспособность при попадании влаги, а также иметь противоскользящие и смягчающие удары элементы на корпусе): требуется. Предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространенных образовательных и общесистемных приложений: требуется.		
1.4	Учебное оборудование			
1.4 .1	Шлем виртуальной реальности	Шлем виртуальной реальности. Наличие контроллеров: 2 шт. Разрешение: не менее 1440 x 1600 на глаз. Встроенные стереонаушники: наличие. Встроенные микрофоны: наличие. Встроенные камеры: не менее 2 шт. Возможность беспроводного использования. Совместимость с ПК - рекомендуется.	ко мп ле кт	1
1.4 .2	Ноутбук виртуальной реальности	Разрешение экрана: не менее 1920x1080 пикселей. Производительность процессора (по тесту PassMark - CPU BenchMark <a href="http://www.cpubenchmark.net/">http://www.cpubenchmark.net/</a> ): не менее 9500 единиц. Производительность графической подсистемы (по тесту PassMark Videocard Bench-mark <a href="http://www.videocardbenchmark.net/">http://www.videocardbenchmark.net/</a> ): не менее 11000 единиц. Объем оперативной памяти: не менее 8 Гб. Объем памяти видеокарты: не менее 6 Гб. Объем твердотельного накопителя: не менее 256 Гб. Наличие русской раскладки клавиатуры: требуется. Наличие цифрового видеовыхода, совместимого с поставляемым шлемом виртуальной реальности: требуется. Предустановленная ОС с графическим пользовательским интерфейсом, обеспечивающая работу распространенных образовательных и общесистемных приложений: требуется.	шт .	1
1.4 .3	Фотограмметрическое программное обеспечение	Программное обеспечение для обработки изображений и определения формы, размеров, положения и иных характеристик объектов на плоскости или в пространстве.	шт .	1
1.4 .4	Квадрокоптер, тип 1	Форм-фактор: устройство или набор для сборки. Канал связи управления квадрокоптером: наличие. Максимальная дальностью передачи данных: не менее 2 км. Бесколлекторные моторы: наличие. Полетный контроллер: наличие. Поддержка оптической системы навигации в помещении: наличие. Модуль фото/видеокамеры разрешением не менее 4К: наличие. Модуль навигации GPS/ГЛОНАСС: наличие. Пульт управления: наличие. Аккумуляторная батарея с зарядным устройством: наличие. Программное приложение для программирования и управления квадрокоптером, в том числе для смартфонов: наличие.	шт .	1
1.4 .5	Квадрокоптер, тип 2	Форм-фактор: устройство или набор для сборки. Канал связи управления квадрокоптером: наличие. Коллекторные моторы: наличие. Полетный контроллер с возможностью программирования: наличие. Поддержка оптической системы навигации в помещении: наличие. Модуль Wi-Fi видеокamеры: наличие. Камера оптического потока: наличие. Аккумуляторная батарея с зарядным устройством: наличие. Программное приложение для программирования и управления квадрокоптером, в том числе для смартфонов.	шт .	3
1.4 .6	Смартфон	Совместимость с квадрокоптером п. 1.4.4 Диагональ экрана: не менее 6.4. Разрешение экрана: не менее 2340x1080 пикселей. Встроенная память: не менее 64 ГБ. Оперативная память: не менее 4 Гб. Емкость аккумулятора: не менее 4000 мАч. Вес: не более 200 гр.	шт .	1
3	Медиазона			
3.1	Фотоаппарат с объективом	Количество эффективных пикселей: не менее 18 млн. Разъем для микрофона 3,5 мм: наличие. Запись видео: наличие.	шт .	1

3.2	Карта памяти для фотоаппарата	Объем памяти: не менее 64 Гб. Класс: не ниже 10. Совместимость с п 3.1.	шт	2
3.3	Штатив	Максимальная нагрузка: не менее 2 кг. Максимальная высота съемки: не менее 148 см.	шт	1
3.4	Микрофон	Длина кабеля: не менее 3 метров. Совместимость с п 3.1. Возможность подключения к ноутбуку/ПК/фотоаппарату: наличие.	шт	1
6	Программное обеспечение, распространяемое бесплатно			
6.1	Программное обеспечение для 3D-моделирования	Облачный инструмент САПР/АСУП, охватывающий весь процесс работы с изделиями — от проектирования до изготовления	лицензия	-
6.2	Программное обеспечение для подготовки 3D-моделей к печати	Инструмент для перевода формата файла из одного типа в другой, понятный 3D-принтеру (п. 1.1.1). Наличие русского языка. Применяется также для масштабирования изделий, расположения на рабочем столе, установки параметров печати и т. д.	лицензия	-

Дидактические материалы: раздаточные материалы, схемы, наглядные пособия, информационная и справочная литература

#### Информационное обеспечение

1. Упражнений на платформе Stepik с теоретическим и практическим материалом языка Python;

2. Интерактивных проверочных заданий в приложениях Kahoot и Learning Apps;

3. Мейер Эрик А. CSS. Карманный справочник. — Вильямс, 2017.

4. Купер Нейт Как создать сайт. Комикс-путеводитель по HTML, CSS и WordPress. — Манн, Иванов, Фербер, 2019.

5. Веру Лиа Секреты CSS. Идеальные решения ежедневных задач. — Питер, 2016

Кадровое обеспечение: преподаватель информатики.

#### 2.3. Формы аттестации.

Аттестация (промежуточная и итоговая) по данной программе в формах, определенных учебным планом.

Оценивание результатов проводится по уровневой системе (высокий, средний, низкий).

#### 2.4. Оценочные материал.

Ф.И. учащегося	ТБ работы с различными инструментами	Умение работать с различными материалами	Умение комбинировать различные материалы	Организованность и самостоятельность обучающихся	Культура поведения и общения учащихся	Дружеские отношения в коллективе

Ключ Критерии оцениваются в диапазоне от 1 до 3 баллов, где:

- 1 балл — низкий уровень освоения программы;
- 2 балла — средний уровень освоения программы;
- 3 балла — высокий уровень освоения программы.

### *2.5. Методические материалы.*

Для реализации данной программы методы обучения выбираются в зависимости от этапа учебного процесса и уровня усвоения знаний различными учащимися.

Методы обучения:

Наглядный практический, самостоятельного изучения, метод стимуляции и поощрения, игровой, словесный

Методы воспитания:

Убеждения, стимулирование, мотивация, поощрение.

Девизом сотворчества с детьми является «Обучая воспитывать — воспитывать обучая». При этом доминирует аксиома: педагог и дети делают одно дело, все заинтересованы сделать его хорошо.

Для реализации данной программы используются следующие педагогические технологии: информационные технологии, технология группового обучения, технология дифференцированного обучения, технология развивающего обучения, технология разноуровневого обучения, технология игровой деятельности, технология коллективной творческой деятельности, здоровьесберегающая технология,

Описание методов обучения:

Словесные методы обучения - словесный (объяснение нового материала, беседа, рассказ и т.д.). Педагогу необходимо как можно меньше объяснять самому, стараться вовлекать детей в обсуждение, нельзя перегружать, торопить детей и сразу стремиться на помощь. Ребенок должен попробовать преодолеть себя, в этом он учится быть взрослым, мастером.

Методы проблемного обучения - Выбирая изделие для изготовления, желательно спланировать объем работы на одно занятие, если времени требуется больше, дети заранее должны знать, какая часть работы останется на второе занятие. Трудные операции, требующие значительного умственного напряжения и мышечной ловкости, обязательно должны быть осознаны детьми как необходимые.

Наглядный метод обучения - наглядные материалы: рисунки, фотографии, таблицы, схемы, чертежи, журналы; демонстрационные материалы: готовые работы, образцы, инструменты и приспособления; демонстрационные видеоматериалы. После изложения теоретических сведений педагог вместе с детьми переходит к практической деятельности. При этом используется для показа интерактивная доска. Таким образом педагог раскрывает творческие возможности работы над определённым заданием.

Игровая гимнастика - игровая гимнастика в виде упражнений (рисунок в воздухе). Чтобы дети быстро не утомлялись и не теряли интерес к предмету, полезно вводить смену видов деятельности и чередование технических

приёмов с игровыми заданиями. Также используются практические задания, выполнение которых предполагает организацию коллективной работы детей.

Формы организации учебного занятия — Беседа, игра, открытое занятие, практическое занятие, выставки,

Структура учебного занятия:

1) Организационная часть (1 минута).

Дети проходят и рассаживаются по своим местам.

2) Вводная часть: Беседа с детьми (5 минут).

Педагог объявляет детям тему, объясняет, как будет проходить занятие.

3) Вводный технологический инструктаж (2 мин).

Педагог рассказывает технологические этапы работы над изделием.

4) Самостоятельная работа, текущий инструктаж педагога (15 минут). Педагог показывает и поэтапно выполняет работу, периодически обходя детей, помогая и указывая на ошибки.

5) Динамическая пауза, физминутка (2 минуты).

Отдых для детей.

6) Подведение итогов. Рефлексия. (4 минут).

Педагог, обсуждая с детьми их работы, задаёт контрольные вопросы и мягко указывает на ошибки и недочёты в работе, отмечает удаchi. 7) Завершение работы (1 минута).

СОГЛАСОВАНО

Протокол заседания  
методического объединения  
учителей математики и физики  
от «31» августа 2021 года  
руководитель МО  
\_\_\_\_\_ Т.А.Червоненко

СОГЛАСОВАНО

Заместитель директора по  
ВР  
\_\_\_\_\_/\_\_\_\_\_  
«31» августа 2021 года