

муниципальное автономное общеобразовательное учреждение  
«Совхозная средняя общеобразовательная школа»

Принята на заседании  
педагогического совета  
Протокол № 22  
от « 30 » 06 2022 г.

Утверждаю  
Директор МАОУ «Совхозная школа»  
С.Н. Галкин  
« 30 » 06 2022 г.



Дополнительная общеобразовательная общеразвивающая программа  
**«Техностарт»**  
технической направленности

Возраст обучающихся: 12 - 14 лет

Срок реализации: 1 год

Составитель: Васильева Евгения Владимировна  
учитель информатики

Совхозный, 2022

## Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Техностарт» составлена в соответствии с действующим законодательством, Уставом и локальными актами учреждения.

**Направленность:** техническая

**Актуальность:** По результатам исследования компании Jet Brains (<https://www.jetbrains.com/ru-ru/lp/devecosystem-2019/>) язык Python – самый изучаемый в 2019 году и один из самых востребованных на рынке труда. Python достаточно широко используется при изучении основ алгоритмизации и программирования в школьном курсе информатики. В частности, используется в качестве базового языка для изучения в УМК по информатике авторов Поляков К. Ю., Еремин Е. А., допущенном к использованию в общеобразовательных организациях Министерством просвещения РФ (Приказ Министерства Просвещения РФ от 28 декабря 2018 года № 345).

Изучение Python в школе откроет ученикам возможности дальнейшего развития в области IT и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решению заданий ЕГЭ.

Программа структурирована на интеграции и опережении с предметами общеобразовательной школы:

- математика (измерение времени, скорости, ускорения и расстояний, с переменными, случайными и пороговыми величинами, изучение геометрических, тригонометрических концепций. Переменная, константа, массив);
- физика (решение задач по механике);
- информатика (алгоритм, принципы программирования, алгоритмическое мышление);

**Отличительной особенностью программы** является подход к организации образовательного процесса. Реализация программы предполагает проведение аудиторных занятий с использованием вариативных дистанционных ресурсов:

- онлайн уроков, размещенных на платформе Stepik;
- интерактивных заданий (опросов, викторин, дидактических игр и др.), размещенных на бесплатных специализированных сервисах Kahoot и Learning Apps;
- проектных заданий для создания учащимися игр на языке Python.

**Уровень сложности программы:** базовый

**Адресат программы:** обучающиеся 7-8 классов

**Планируемое количество учащихся по годам обучения:** обучающиеся занимаются в группах по 10-12 человек.

**Срок освоения программы:** 1 год

**Режим занятий:** 1 раз в неделю по 2 часа

**Часовой объем программы:** 72 часа

**Форма обучения:** смешанная: очное и онлайн-обучение. В очных занятиях в группах по 10-15 человек используются задания на онлайн-платформе с автоматизированной

проверкой. Задания на платформе доступны ученикам для самостоятельного изучения в любое время.

В ходе курса запланирован плавный переход к формату “перевернутого класса”: на первых пяти занятиях учеников знакомят с онлайн-платформой и основными инструментами программирования на Python. Начиная с шестого занятия предлагаются различные формы самостоятельной активности на уроке, с групповой взаимопроверкой. После десятого урока иногда ученикам предлагается самостоятельно изучить новую тему, опираясь на конспект онлайн-курса, а роль учителя сдвигается к консультации и модерации. Последние пять уроков отведены на почти самостоятельную работу над проектами.

**Формы организации образовательного процесса:** все практические занятия выполняются в виде индивидуальных или групповых проектов и решении задач

**Цель программы:** сформировать у обучающихся навыки программирования на языке высокого уровня Python

**Задачи программы:**

- 1) Научить приемам программирования на языке Python и применять полученные знания для решения практических задач.
- 2) Обеспечить возможность приобретения обучающимися практических навыков программирования, отладки и выполнения на компьютере конкретных задач.

**Планируемые результаты:**

#### **Личностные**

По окончании курса учащийся сможет:

- Программировать на языке Python.
- Использовать инструменты разработки среды Wing.
- Самостоятельно реализовывать проекты, связанные с разработкой игр.

#### **Метапредметные**

- Ставить учебные цели.
- Формулировать достигнутый результат.
- Планировать свою самостоятельную учебно-познавательную деятельность; выбирать индивидуальную траекторию достижения учебной цели.
- Определять подходы и методы для достижения поставленной цели.
- Отбирать необходимые средства для достижения поставленной цели.
- Осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности.
- Проводить рефлексию своей учебно-познавательной деятельности.

#### **Предметные:**

Практическим результатом работы служит финальный проект каждого ученика: 2D игра на движке Pygame, либо серия самостоятельно разработанных мини-проектов в консольном режиме языка Python.

**Формы контроля (аттестации):** Аттестация проводится в форме выполнения индивидуальных и групповых заданий по пройденному материалу. Контроль в указанной

форме осуществляется как промежуточный, так и итоговый. Отметочная форма контроля отсутствуют.

Для заданий на онлайн-тренажере указан необходимый минимум — 70% выполненных заданий, чтобы тема считалась пройденной успешно и был открыт доступ к следующей теме.

После каждой темы в онлайн-курсе стоит итоговая работа: от ученика требуется в ограниченное время (три часа) решить набор задач по пройденной теме. В среднем, ученик справляется с решением за 30 минут. Преподаватели могут использовать эти итоговые работы в качестве промежуточных проверочных работ.

В конце курса, по итогам работы над групповыми и индивидуальными проектами проводится обсуждение результатов в коллективе с опорой на чек-лист, исправление ошибок и, тем самым, коррекция и закрепление полученных знаний.

## Учебный план

№ п/п	Разделы и темы	Всего часов	В том числе:		Формы контроля
			теория	практика	
1	Знакомство с Python	2	1	1	
2	Ввод-вывод данных	6	3	3	Решение задач
3	Условный оператор	6	3	3	Интерактивные упражнения
4	Типы данных	4	1	3	Краткий опрос, контрольная работа на платформе
5	Циклы	12	4	8	Краткий опрос, контрольная работа на платформе
6	Строки	8	2	6	Краткий опрос, контрольная работа на платформе
7	Списки	8	2	6	Краткий опрос, контрольная работа на платформе
8	Функции	10	4	6	Краткий опрос, контрольная работа на платформе
9	Работа над мини-проектами	14	2	12	Защита проектов
10	Заключительное занятие	2		2	Олимпиада
<b>ИТОГО:</b>		72	22	50	

## Содержание программы

### **Знакомство с Python.**

*Теоретическая часть.* Новейшие достижения науки и техники в смежных областях. История развития языков программирования. Техника безопасности.

*Практическая часть.* Установка среды программирования. Регистрация на платформе. Подключение к курсу.

### **Раздел «Ввод-вывод данных»**

*Теоретическая часть* Команды Input() и Print(). Параметры sep, end. Переменные. Комментарии. PEP 8. Работа с целыми числами.

*Практическая часть.* Отработка навыков работы с переменными, использования комментариев в коде. Отработка операций с целыми числами. Обработка цифр числа.

### **Раздел «Условный оператор»**

*Теоретическая часть.* Условный оператор. Логические операции and, or, not. Вложенный и каскадный условный оператор.

*Практическая часть.* Решение задач с использованием условного оператора и логических операций. Решение задач с использованием каскадного условного оператора и вложенных ветвлений. Рефлексия “Повар”

### **Раздел «Типы данных»**

*Теоретическая часть.* Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in.

*Практическая часть.* Групповая работа над презентацией. Самостоятельное изучение темы в малых группах и создание презентации. Решение задач на платформе.

### **Раздел «Циклы»**

*Теоретическая часть.* Цикл for. Функция range(). Частые сценарии при написании циклов. Расширенные операторы присваивания. Цикл с предусловием while. Операторы break, continue, else. Вложенные циклы.

*Практическая часть.* Решение задач с использованием различных видов цикла. Подвижная игра с функцией range(). Самостоятельное изучение новой темы в командах. Решение задач на платформе на отработку частых сценариев.

### **Раздел «Строки»**

*Теоретическая часть.* Строковый тип данных: индексация и срезы. Методы строк.

*Практическая часть.* Решение задач на платформе: использование методов строк.

### **Раздел «Списки»**

*Теоретическая часть.* Основы работы со списками. Методы списков. Вывод элементов списка. Строковые методы split() и join(). Списочные выражения.

*Практическая часть.* Решение задач на платформе: сначала всех пропущенных, потом - на использование списков. Соревновательное подведение итогов изучения темы “Списочные выражения”. Рефлексия командной работы.

### **Раздел «Функции»**

*Теоретическая часть.* Функции. Локальные и глобальные переменные. Функции, возвращающие значения.

*Практическая часть.* Решение задач с использованием функций. Игра на введение в проектную деятельность.

### **Раздел «Работа над мини-проектами»**

*Теоретическая часть.* Проектный подход к заданиям. Принципы краткой презентации.

*Практическая часть.* Работа над общим проектом-образцом на платформе. Самостоятельная работа над проектами. Выступление с презентацией собственного проекта.

### **Условия реализации программы**

1. Компьютерный класс в расчете 1 компьютер на 1 ребенка
2. Язык программирования Python версии не ниже 3.0, установленный на каждом компьютере
3. Браузер с выходом в интернет со скоростью не менее 6 Мбит/с на каждом компьютере

### **Методическое обеспечение**

Методические материалы курса состоят из:

1. Методических указаний для учителя в текстовом виде;
2. Презентации с иллюстративным изложением теоретического материала;
3. Упражнений на платформе Stepik с теоретическим и практическим материалом языка Python;
4. Интерактивных проверочных заданий в приложениях Kahoot и Learning Apps;
5. Подвижных игр, направленных на закрепление знаний, полученных на занятии.
6. Инструкций для проведения рефлексии процесса обучения с учениками.



### Календарный учебный график

Месяц	№ недели	Год обучения, форма занятия		
		Т	П	К
Сентябрь	1	Т	П	
	2	Т	П	
	3	Т	П	
	4	Т	П	К
Октябрь	1	Т	П	
	2	Т	П	
	3	Т	П	К
	4	Т	П	
Ноябрь	1	Т	П	
	2	Т	П	
	3	Т	П	
	4	Т	П	
Декабрь	1	Т	П	
	2	Т	П	К
	3	Т	П	
	4	Т	П	К
Январь	1	Т	П	
	2	Т	П	
	3	Т	П	
	4	Т	П	
Февраль	1	Т	П	К
	2	Т	П	
	3	Т	П	
	4	Т	П	
Март	1	Т	П	
	2	Т	П	
	3	Т	П	
	4	Т	П	К
Апрель	1	Т	П	
	2	Т	П	
	3	Т	П	
	4	Т	П	
Май	1		П	
	2		П	К
	3		П	
	4		П	К
<b>Всего часов</b>		<b>32</b>	<b>36</b>	<b>8</b>
<b>Итого за год</b>	<b>36 недель, 72 часа</b>			

## Календарный план воспитательной работы

№ п/п	Форма и название мероприятия	Сроки проведения (указать месяц)
<b>Направление 1. Формирование и развитие творческих способностей учащихся, выявление и поддержка талантливых учащихся</b>		
1	<i>Районная научно-практическая конференция «Исследователи XXI века»</i>	Март
2	<i>Районный конкурс «Пьедестал успеха – 2022»</i>	Май
3	<i>Школьный и муниципальный тур олимпиады школьников по информатике</i>	Октябрь - декабрь
<b>Направление 2. Духовно-нравственное, гражданско-патриотическое воспитание, формирование общей культуры учащихся, профилактики экстремизма и радикализма</b>		
1	<i>Международный день солидарности в борьбе с терроризмом</i>	Сентябрь
2	<i>Участие в акции «Окна победы»</i>	Май
<b>Направление 3. Социализация, самоопределение и профессиональная ориентация учащихся</b>		
1	<i>Экскурсия в Кванториум, IT_Куб г.Ижевск</i>	Март
2	<i>Встреча с выпускниками школы, работающими программистами</i>	Ноябрь
3	<i>Всероссийский образовательный проект «Урок цифры»</i>	Октябрь-май
<b>Направление 4. Формирование культуры здорового и безопасного образа жизни и комплексной профилактической работы</b>		
1	<i>Проведение мастер-класса по здоровому образу жизни</i>	Ноябрь
2	<i>Проведение инструктажей по ПБ, ТБ в здании, на занятиях</i>	Сентябрь, январь
3	<i>Участие в акции «Нет наркотикам»</i>	Апрель

## Контрольно-измерительные материалы

1. Какую последовательность чисел даст вам вызов функции `range(6)`?
  - a. 1, 2, 3, 4, 5, 6
  - b. 6
  - c. 1, 2, 3, 4, 5
  - d. 0, 1, 2, 3, 4, 5, 6
  - e. 0, 1, 2, 3, 4, 5 ✓
2. Какую последовательность чисел даст вам вызов функции `range(2, 6)`?
  - a. 2, 3, 4, 5, 6
  - b. 1, 2, 3, 4, 5
  - c. 2, 3, 4, 5 ✓
  - d. 2, 6
3. Какую последовательность чисел даст вам вызов функции `range(10, 5, -1)`?
  - a. 10, 9, 8, 7, 6, 5
  - b. 10, 5, -1
  - c. 10, 5, 0
  - d. 10, 9, 8, 7, 6 ✓
4. Определите, что выведет следующий фрагмент кода?
  - a) 15 раз Python awesome! на каждой строке ✓
  - b) 25 раз Python awesome! на каждой строке
  - c) 10 раз Python awesome! на каждой строке
  - d) 15 раз Python awesome! на одной строке через пробел
5. Определите какую задачу решает следующий фрагмент кода:

```
n = int(input())
counter = 0
for i in range(1, n + 1):
    if i % 3 == 0 and i % 7 != 0:
        counter += 1
print(counter)
```

- a) выводит сумму чисел от 1 до n-1 кратных 7, но не кратных 3
  - b) выводит количество чисел от 1 до n-1 кратных 3, но не кратных 7
  - c) выводит количество чисел от 1 до n кратных 3, но не кратных 7 ✓
  - d) выводит сумму чисел от 1 до n кратных 3, но не кратных 7
6. Соберите программу, вычисляющую сумму чисел от 1 до введенного натурального числа N. Расположите элементы в правильном порядке.

```
for i in range(1, n + 1):      3
print (total)                 5
total += i                    4
total = 0                     2
n = int(input())              1
```

7. Какое число нужно написать вместо многоточия, чтобы цикл выполнялся ровно 7 раз?

```
i = ...  
while i <= 10:  
    print ('Python!')  
    i += 1
```

Ответ: 4

8. Определите какую задачу решает следующий фрагмент кода:

```
n = int(input())  
res = 1  
i = 2  
while i <= n:  
    res *= i  
    i += 1  
print(res)
```

- a) выводит 2 в степени n  
b) выводит сумму чисел от 1 до n  
c) выводит сумму чисел от 2 до n  
d) выводит факториал числа n ✓

9. Определите какую задачу решает следующий фрагмент кода:

```
n = int(input())  
i = 2  
while n % i != 0:  
    i += 1  
print(i)
```

- a) выводит количество цифр в числе  
b) выводит сколько раз делится введенное число на 2  
c) выводит максимальный делитель числа  
d) выводит минимальный делитель числа, отличный от единицы ✓

10. Соберите программу, вычисляющую количество цифр введенного натурального числа. Расположите строки в правильном порядке

n = int(input())	1
print(counter)	6
counter += 1	4
while n > 0:	3
counter = 0	2
n //= 10	5

**Критерии оценивания:** 8-10 правильных ответов – высокий уровень, 5-7 правильных ответов – средний уровень, 0-4 правильных ответа – низкий уровень.

## **СПИСОК ЛИТЕРАТУРЫ ДЛЯ ПЕДАГОГА**

1. Васильев, А. Н. Python на примерах: практ. курс /А. Н. Васильев - Наука и Техника, 2019 - 432 с.
2. Гэддис, Т. Начинаем программировать на Python: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
3. Прохоренок, Н. А. Python 3: самое необходимое: практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
4. Седжвик, Р. Программирование на языке Python /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
5. Харрисон, М. Как устроен Python: практ. курс / М. Харрисон - Питер, 2002 - 272 с.

## **СПИСОК ЛИТЕРАТУРЫ ДЛЯ ОБУЧАЮЩИХСЯ**

Не предусмотрен

## **ЭЛЕКТРОННЫЕ РЕСУРСЫ**

1. Курс Поколение Python: курс для начинающих на платформе Stepik. (<https://stepik.org/course/58852/syllabus>)