

**Муниципальное бюджетное общеобразовательное учреждение  
«Средняя общеобразовательная школа п.Степное Калининского района Саратовской  
области»**

ПРИНЯТО

заседанием педагогического совета

Протокол № 1 от 31.08.2022 г.

УТВЕРЖДАЮ

И.о.директора МБОУ «СОШ п.Степное  
Калининского района Саратовской области»

 /В.С. Саخنo/

Приказ № 116 от 01.09.2022 г

**Дополнительная общеобразовательная  
общеразвивающая программа  
«Программирование. Python.»**

Направленность: техническая  
Возраст обучающихся: 15-18 лет  
Срок реализации: 1 год

Автор-составитель:  
Саخنo Владислав Степанович,  
педагог дополнительного образования

п. Степное  
2022 год

# 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

## Пояснительная записка

Дополнительная общеобразовательная, общеразвивающая программа «Программирование. Pyhton.» является модифицированной и относится к технической направленности. Программа разработана на основании:

1. Федерального закона РФ от 29.12.2012 г. № 273 – ФЗ «Об образовании в Российской Федерации».
2. «Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (утв. Приказом Министерства Просвещения Российской Федерации от 9 ноября 2018 г. № 196, с изменениями от 30.09.2020 г.)
3. Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)» (утв. Письмом Министерства образования и науки РФ от 18.11.15 № 09-3242).
4. «Правил персонифицированного дополнительного образования в Саратовской области» (утв. Приказом Министерства образования Саратовской области от 21.05.2019г. № 1077, с изменениями от 14.02.2020 года, от 12.08.2020 года).
5. Положения о разработке и условиях реализации дополнительных общеобразовательных программ МБОУ «СОШ п.Степное Калининского района Саратовской области».
6. Санитарных правил 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (утв. Постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28).

В современную жизнь человека все шире внедряются компьютеры и информационные технологии. Поэтому все большее значение приобретает компьютерная грамотность.

Программа изучения компьютерной грамотности состоит из двух разделов: пользовательского курса и программирования. Раздел «Программирование» в школьном курсе представлен языком программирования Pascal, а многим учащимся хочется познакомиться с другими языками программирования, самим попробовать разработать программы, которые можно использовать на уроках и во внеурочной деятельности. Данная Программа позволяет реализовать эти желания, так как уделяется большое внимание практической работе учащихся на компьютере, самостоятельной разработке ими программ для решения практических задач.

### 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.

**Объем программы:** 144 часа.

**Срок реализации программы:** 1 год

**Форма обучения:** очная.

**Особенности набора в группы:** прием детей в группы свободный по сертификату дополнительного образования.

**Количество обучающихся в группе:** 12 – 15 человек.

**Режим занятий:** 2 раза в неделю по 2 часа (2x40 минут).

**Адресат программы:** обучающиеся в возрасте 15 – 18 лет.

#### **Возрастные особенности обучающихся 15 – 18 лет.**

Старший школьный возраст — 15-17 лет (ранняя юность). Главное психологическое приобретение ранней юности — это открытие своего внутреннего мира, внутреннее «Я». Главным измерением времени в самосознании является будущее, к которому он (она) себя готовит. Ведущая деятельность в этом возрасте — учебно-профессиональная, в процессе которой формируются такие новообразования, как мировоззрение, профессиональные интересы, самосознание, мечта и идеалы

В этом возрасте учащиеся стремятся утвердиться в своей жизненной позиции и программа создает условия для развития личности подростка, их желанием заявить об этом.

#### **Цели и задачи программы**

**Цель программы:** обучение учащихся программированию посредством языка Python, развитие инженерного мышления, воспитание конкурентно способной личности.

#### **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.

**Планируемые результаты освоения программы**

Программой предусматривается достижение обучающимися предметных, метапредметных и личностных результатов.

**Предметные:**

- владение основами разработки алгоритмов и составления программ на Python;
- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу
- основы дизайнерского оформления созданных программ.;

**Метапредметные:**

- овладение навыками самостоятельного приобретения новых знаний
- планирование учебной деятельности, самоконтроль и оценка результатов своей деятельности
- приобретение опыта самостоятельного поиска, анализа и отбора информации с использованием различных источников и новых информационных технологий для решения задач

**Личностные:**

- сформированность познавательных интересов, интеллектуальных и творческих способностей учащихся
- умение осуществлять совместную информационную деятельность, в частности при выполнении учебных проектов;
- готовность к выбору жизненного пути в соответствии с собственными интересами и возможностями

**Содержание программы  
Учебный план.**

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
<b>1. Введение в Python</b>		<b>14</b>	<b>6</b>	<b>8</b>	
1.1	Язык Python. Среда программирования. Особенности ввода-вывода	4	2	2	Практическое задание
1.2	Типы данных, операции.	4	2	2	Выполнение

	Оператор присваивания				упражнений
1.3	Числа. Стандартные операции	6	2	4	Практическое задание
<b>2. Алгоритмические инструкции</b>		<b>22</b>	<b>6</b>	<b>16</b>	
2.1.	Условный оператор	2	2	-	Опрос
2.2.	Цикл while. Теория чисел	10	2	8	Практическое задание
2.3.	Цикл for	10	2	8	Практическое задание
<b>3. Строки</b>		<b>20</b>	<b>8</b>	<b>12</b>	
3.1.	Литералы строк	4	2	2	Практическое задание
3.2.	Срезы строк	4	2	2	Практическое задание
3.3.	Методы строк	12	4	8	Практическое задание
<b>4. Функции</b>		<b>20</b>	<b>8</b>	<b>12</b>	
4.1.	Парадигма и преимущества структурного программирования	2	2	-	Опрос
4.2.	Граф и стек вызовов функций. Области видимости переменных	4	2	2	Практическое задание
4.3.	Прямая рекурсия	10	2	8	Практическое задание
4.4.	Косвенная рекурсия	4	2	2	Решение задач повышенной трудности
<b>5. Списки и кортежи</b>		<b>20</b>	<b>8</b>	<b>12</b>	
5.1.	Списки и кортежи в Python. Сходства и различия	2	2	-	Опрос
5.2.	Операции со списками	10	2	8	Практическое задание
5.3.	Срезы списков	4	2	2	Практическое задание
5.4.	Матрицы. Операции над матрицами	4	2	2	Решение задач повышенной трудности
<b>6. Словари и множества</b>		<b>8</b>	<b>4</b>	<b>4</b>	
6.1.	Словари	4	2	2	Практическое задание
6.2.	Множества	4	2	2	Практическое задание
<b>7. Объектно-ориентированное программирование</b>		<b>38</b>	<b>4</b>	<b>34</b>	
7.1.	Классы в Python	2	2	-	Опрос
7.2	Разработка собственного класса	6	2	4	Практическое задание
7.3.	Разработка и программирование собственного проекта	30	-	30	
<b>8. Заключительное занятие</b>		<b>2</b>	<b>-</b>	<b>2</b>	
8.1.	Подведение итогов. Индивидуальный проект.	2	2		Защита проекта

	<b>Всего</b>	<b>144</b>	<b>44</b>	<b>100</b>	

## СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

### 1. Введение в Python

*Основные понятия:* трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

#### **Язык Python. Среда программирования. Особенности ввода-вывода**

*Теория.* Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы.

*Практика.* Установка языка программирования Python 3.8 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

#### **Типы данных, операции. Оператор присваивания**

*Теория.* Ввод и вывод числовой информации.

*Практика.* Тренировочное задание на ввод и вывод числовой информации.

#### **Числа. Стандартные операции**

*Теория.* Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).

*Практика.* Решение простых задач в интерактивном режиме.

### 2. Алгоритмические инструкции

*Основные понятия:* логический тип данных, логические операции (and, or, not A), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

#### **Условный оператор**

*Теория.* Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

#### **Цикл while**

*Теория.* Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

*Практика.* Решение задач на анализ чисел потока и целочисленной арифметики.

#### **Цикл for**

*Теория.* Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

*Практика.* Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

### 3. Строки

*Основные понятия:* символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные

escape- последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

### **Литералы строк**

*Теория.* Понятие «литералы строк».

*Практика.* Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

### **Срезы строк**

*Теория.* Форматирование строки.

*Практика.* Преобразование строки. Решение упражнений.

### **Методы строк**

*Теория.* Методы работы со строкой.

*Практика.* Применение методов строки. Решение задач.

## **4. Функции**

*Основные понятия:* подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

### **Парадигма и преимущества структурного программирования**

*Теория.* Обзор парадигм программирования. Особенности применения языков программирования.

### **Граф и стек вызовов функций. Области видимости переменных**

*Теория.* Применение стек и граф вызовов функций.

*Практика.* Выполнение тренировочных упражнений.

### **Прямая рекурсия**

*Теория.* Понятие прямой рекурсии и ее применение.

*Практика.* Выполнение тренировочных упражнений.

### **Косвенная рекурсия**

*Теория.* Понятие косвенной рекурсии и ее применение.

*Практика.* Выполнение тренировочных упражнений.

*Практические занятия к темам 4.2., 4.3., 4.4.:* Нахождение суммы чисел. Числа Фибоначчи. Вычисление степени. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности.

## **5. Списки и кортежи**

*Основные понятия:* список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

### **Списки и кортежи в Python. Сходства и различия**

*Теория.* Представление списка и кортежа в памяти компьютера, сходства и различия.

### **Операции со списками**

**Теория.** Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

**Практика.** Решение задач на ввод-вывод элементов одномерного и многомерного списка.

### **Срезы списков**

**Теория.** Методы работы со списком и кортежем. Методы сортировки списка.

**Практика.** Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

### **Матрицы. Операции над матрицами**

**Теория.** Вычисление сложности алгоритма. Многомерные списки.

**Практика.** Решение задач повышенной трудности.

## **6. Словари и множества**

Основные понятия: *словарь, множество, ключ, кодирование.*

### **Словари**

**Теория.** Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

**Практика.** Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

### **Множества**

**Теория.** Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

**Практика.** Решение задач повышенной трудности.

## **7. Объектно-ориентированное программирование (ООП)**

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

### **Классы в Python**

**Теория.** Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

### **Разработка собственного класса**

**Теория.** Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

**Практика.** Создание собственного класса.

### **Разработка и программирование собственного проекта**

**Практика.** Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.

## **8. Подведение итогов**

### **Защита индивидуального проекта**



## 1.5. Формы аттестации планируемых результатов и их периодичность.

Для отслеживания результативности образовательного процесса используются следующие виды контроля:

- текущий контроль (осуществляться по результатам выполнения учащимися практических заданий);
- промежуточный контроль (выполнение самостоятельных и зачетных работ);
- итоговый контроль (защита проектов)

*Критерии оценивания практического задания:*

- учащийся набирает 10 баллов, если он показывает отличное понимание теории и применение её на практике, программу пишет правильно, программа работоспособна, приводит к нужному результату;
  - учащийся набирает 9 баллов, если он показывает достаточно полное понимание теории и применение её на практике, программу пишет правильно, допустимо одна неточность или ошибка, которую исправляет сам, программа работоспособна, приводит к правильному результату;
  - учащийся набирает 8 баллов, если он показывает достаточно полное понимание теории и применение её на практике, программу пишет правильно, допустимо две неточности или ошибки, которые исправляет сам, программа работоспособна, приводит (после исправления) к правильному результату;
  - учащийся набирает 7 баллов, если он показывает достаточно полное понимание теории и применение её на практике, программу пишет правильно, допустимо три неточности или ошибок, которые исправляет сам, программа работоспособна, приводит (после исправления) к правильному результату;
  - учащийся набирает 6 баллов, если он показывает приемлемое понимание теории и удовлетворительное применение её на практике, программу пишет в целом правильно, допустимо четыре неточности или ошибок, которые исправляет самостоятельно по требованию учителя, программа работоспособна, приводит (после исправления) к правильному результату;
  - учащийся набирает 5 баллов, если он показывает приемлемое понимание теории и удовлетворительное применение её на практике, программу пишет в целом правильно, допускает пять неточностей или ошибок в сумме, которые исправляет самостоятельно по требованию учителя, программа при этом работоспособна, приводит (после исправления) к правильному результату;
  - если же учащийся, показывает неудовлетворительное понимание теории и применение её на практике, в программе допускает шесть и более ошибок, не может их исправить по требованию учителя, то он получает ниже пяти баллов.
- отметка «5»:если учащийся набрал 9 - 10 баллов;

- отметка «4»: если учащийся набрал 7 - 8 баллов ;
- отметка «3»: если учащийся набрал 5 - 6 баллов ;
- отметка «2»: если учащийся набрал ниже 5 баллов.

*Критерии оценивания самостоятельных работ:*

- оценка «5» ставится если выполнено 85-100% всей работы;
- оценка «4» ставится, если выполнено 65-84% всей работы;
- оценка «3» ставится, если выполнено 40-64% всей работы;
- оценка «2» ставится, если выполнено менее 40% всей работы.

Итоговая практико-значимая работа рассматривается как обобщение опыта усвоения данного курса, систематизирует знания, практические умения и навыки, способы творческой деятельности, полученные в ходе практических занятий, выполнения самостоятельных работ. Итоговая практико-значимая работа представляется в форме законченного проекта.

## **2. Комплекс организационно-педагогические условия.**

### **Методическое обеспечение.**

Программа предусматривает поэтапное ознакомление обучающихся с языком Python, с программированием по принципу "от простого к сложному", от элементарных до больших программ самостоятельной разработки.

**Формы организации занятий:** мини – проекты, самостоятельная работа, творческие задания,

**Основные методы обучения,** применяемые при реализации программы.

Наглядный, проблемный, частично-поисковый, исследовательский, проектный.

### **Формы подведения итога реализации программы**

- защита итоговых проектов;

### **Условия реализации программы**

### **Материально - техническое оснащение занятий:**

Для качественной реализации программного материала необходимо:

- кабинет «Точки роста» для проведения теоретических и практических занятий, с типовой мебелью.
- АРМ педагога (компьютер, проектор);
- рабочие компьютеры учащихся для работы с доступом в Интернет;
- принтер для распечатки заданий.

*Дидактический материал,* необходимый для проведения занятий:

- конспекты занятий;
- распечатки заданий для практикумов;
- презентационные материалы для объяснения;
- карточки с индивидуальными заданиями.

### 2.3 Календарный учебный план

№ п/п	Дата	Время проведения занятий	Форма занятий	Тема	Количество часов	Место проведения	Форма контроля
				<b>1. Введение в Python</b>	<b>14</b>		
1			Комбинированный урок: слушают объяснения учителя, наблюдают за демонстрациями учителя, отвечают на контрольные вопросы.	Язык Python. Среда программирования. Особенности ввода-вывода	1	Кабинет точки роста	Устный опрос
2			Комбинированный	Язык Python. Среда программирования. Особенности ввода-вывода	1	Кабинет точки роста	Практическое задание
3			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
4			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа

5			лекция	Типы данных, операции. Оператор присваивания	1	Кабинет точки роста	тест
6			Комбинированный	Типы данных, операции. Оператор присваивания	1	Кабинет точки роста	Устный фронтальный опрос. Самостоятельная работа
7			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
8			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
9			Комбинированный	Числа. Стандартные операции	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
10			Комбинированный	Числа. Стандартные операции	1	Кабинет точки роста	Устный опрос
11			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
12			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
13			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
14			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
				<b>2.Алгоритмические инструкции</b>	<b>22</b>		Устный фронтальный опрос. Практическая работа
15			Комбинированный	Условный оператор	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
16			Комбинированный	Условный оператор	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа

17			Комбинированный	Цикл while. Теория чисел	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
18			Комбинированный	Цикл while. Теория чисел	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
19			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
20			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
21			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
22			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
23			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
24			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
25			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
26			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
27			Комбинированный	Цикл for	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
28			Комбинированный	Цикл for	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
29			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
30			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа

31			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
32			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
33			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
34			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
35			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
36			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
				<b>3. Строки</b>	<b>20</b>		
37			Лекция	Литералы строк	1	Кабинет точки роста	Устный опрос, фронтальный
38			Комбинированный	Литералы строк	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
39			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
40			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
41			Лекция	Срезы строк	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
42			Комбинированный	Срезы строк	1	Кабинет точки роста	Устный фронтальный опрос. Практическая работа
43			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
44			практикум по	Практика	1	Кабинет	Самостоятельная

			программированию			точки роста	работа
45			Комбинированный	Методы строк	1	Кабинет точки роста	Самостоятельная работа
46			Комбинированный	Методы строк	1	Кабинет точки роста	Устный опрос, фронтальный
47			Комбинированный	Методы строк	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
48			Комбинированный	Методы строк	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
49			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
50			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
51			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
52			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
53			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
54			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
55			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по

							программированию
56			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
				<b>4.Функции</b>	<b>20</b>		
57			Комбинированный	Парадигма и преимущества структурного программирования	1	Кабинет точки роста	Устный опрос, фронтальный
58			Комбинированный	Парадигма и преимущества структурного программирования	1	Кабинет точки роста	Устный опрос, фронтальный
59			Комбинированный	Граф и стек вызовов функций. Области видимости переменных.	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
60			Комбинированный	Граф и стек вызовов функций. Области видимости переменных.	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
61			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
62			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
63			Комбинированный	Прямая рекурсия	1	Кабинет точки роста	Практическая работа по программированию
64			Комбинированный	Прямая рекурсия	1	Кабинет точки роста	Практическая работа по программированию
65			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по



							программированию
66			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
67			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
68			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
69			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
70			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
71			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
72			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
73			лекция	Косвенная рекурсия	1	Кабинет точки роста	Устный опрос, фронтальный
74			Комбинированный	Косвенная рекурсия	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
75			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
76			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа

				<b>5. Списки и кортежи</b>	<b>20</b>		
77			лекция	Списки и кортежи в Python. Сходства и различия	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
78			Комбинированный	Списки и кортежи в Python. Сходства и различия	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
79			лекция	Операции со списками	1	Кабинет точки роста	тест
80			Комбинированный	Операции со списками	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
81			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
82			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
83			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
84			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
85			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
86			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
87			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию

88			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
89			лекция	Срезы списков	1	Кабинет точки роста	тест
90			Комбинированный	Срезы списков	1	Кабинет точки роста	тест
91			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
92			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
93			Комбинированный	Матрицы. Операции над матрицами	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
94			Комбинированный	Матрицы. Операции над матрицами	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
95			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
96			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
				<b>6. Словари и множества</b>	<b>8</b>		
97			Комбинированный	Словари	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
98			Комбинированный	Словари	1	Кабинет точки роста	Устный опрос, фронтальный,

							решение задач
99			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
100			практикум по программированию	Практика	1	Кабинет точки роста	Самостоятельная работа
101			Комбинированный	Множества	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
102			Комбинированный	Множества	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
103			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
104			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
				<b>7. Объектно-ориентированное программирование</b>	<b>38</b>		
105			Комбинированный	Классы в Python	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
106			Комбинированный	Классы в Python	1	Кабинет точки роста	тест
107			Комбинированный	Разработка собственного класса	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач

108			Комбинированный	Разработка собственного класса	1	Кабинет точки роста	Устный опрос, фронтальный, решение задач
109			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
110			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
111			практикум по программированию	Практика	1	Кабинет точки роста	Практическая работа по программированию
112			практикум по программированию	Практика	1		Практическая работа по программированию
113			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Практическая работа по программированию
114			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Практическая работа по программированию
115			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Практическая работа по программированию
116			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Практическая работа по программированию
117			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Практическая работа по программированию
118			Работа над проектом	Разработка и программирование	1	Кабинет точки роста	Практическая работа по





				собственного проекта			
141			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Самостоятельная работа
142			Работа над проектом	Разработка и программирование собственного проекта	1	Кабинет точки роста	Самостоятельная работа
				<b>8.Заключительное занятие</b>	2		
143			Итоговое занятие	Подведение итогов. Индивидуальный проект.	1	Кабинет точки роста	Защита проекта
144			Итоговое занятие	Подведение итогов. Индивидуальный проект.	1	Кабинет точки роста	Защита проекта



## 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, самостоятельных работ и практических заданий;
- итоговый контроль – защита проекта.

## Список литературы

### Для педагога:

1. Лутц М. Изучаем Python, 4 издание / М. Лутц - СПб.: Символ-Плюс, 2011. - 1280 с.
2. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014. - 338с.
3. Н.Прохоренок. В.Дронов Python 3 PyQt 5/ СПб ; БХВ-Петербург, 2016
4. Златопольский Д.М. Основы программирования на Python/ ДМК Пресс, 2017. -284с
5. Учим Python, делая крутые игры / Эл Свейгард; - Москва : Эксмо, 2018. -416с

### Для обучающихся:

1. Брайсон Пэйн Python для детей и взрослых / издательство Бомбора, Москва 2021г
2. Н.Прохоренок. В.Дронов Python 3 Самое необходимое / СПб ; "БХВ-Петербург", 2016
3. Б.Любанович Простой Python/ Санк-Петербург Москва Екатеринбург Воронеж Нижний Новгород Ростов на Дону Самара Минск 2021
4. Наоми Седер Python Экспресс - курс. - СПб.: Питер, 2018.
5. Джейми Чан Python:быстрый старт. - СПб.: Питер, 2021.

### Цифровые образовательные ресурсы

1. [Введение | Основы языка Python \(hexlet.io\)](https://hexlet.io)
2. [python.pdf \(rus-linux.net\)](https://rus-linux.net/python.pdf)

## Приложение №1

1. Запишите значение переменной  $t$ , полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
алг нач цел $t, i$ $t := 2$ нц для $i$ от 1 до 3 $t := t * i$ кц вывод $t$ кон	<pre> DIM i,t AS INTEGER t = 2 FOR i = 1 TO 3 t = t * i NEXT i PRINT t                     </pre>	<pre> Var t,i: integer; Begin t := 2; For i := 1 to 3 do t :=t * i; Writeln(t); End.                     </pre>

Ответ:

2. У исполнителя Квадратор две команды, которым присвоены номера: 1. возведи в квадрат 2. прибавь 3.

Первая из них возводит число на экране во вторую степень, вторая — прибавляет к числу 3. Составьте алгоритм получения из числа 5 числа 127, содержащий не более 5 команд. В ответе запишите только номера команд. (Например, 12212 — это алгоритм: возведи в квадрат прибавь 3 прибавь 3 возведи в квадрат прибавь 3, который преобразует число 2 в число 103).

Если таких алгоритмов более одного, то запишите любой из них.

Ответ:

3. Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала вычисляется длина исходной цепочки символов; если она чётна, то в середину цепочки символов добавляется символ А, а если нечётна, то в начало цепочки добавляется символ Б. В полученной цепочке символов каждая буква заменяется буквой, следующей за ней в русском алфавите (А — на Б, Б — на В и т. д., а Я — на А). Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной была цепочка **ВРМ**, то результатом работы алгоритма будет цепочка **ВГСН**, а если исходной была цепочка **ПД**, то результатом работы алгоритма будет цепочка **РБЕ**.

Дана цепочка символов **ТОР**. Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (т. е. применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)? Русский алфавит: АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ.

Ответ:

4. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6. Количество чисел не превышает 1000. Введённые числа не превышают 30000. Программа должна вывести одно число — количество чисел, оканчивающихся на 6.

Пример работы программы:

Входные данные	Выходные данные
3,12,16,26,30,36,40	3