

Министерство образования Кировской области

КОГОБУ СШ с УИОП г. Нолинска

РАССМОТРЕНО

на педагогическом совете

ул. (Е. Н. Губина)

Протокол № 1 от 28 августа 2024 г.

УТВЕРЖДЕНО

Лешин

Приказ № 21 от 28 августа 2024 г.

Дополнительная общеобразовательная общеразвивающая программа  
технической направленности

РАССМОТРЕНО

**«Программирование на Python»**

УТВЕРЖДЕНО

на педагогическом совете Возраст обучающихся: 16–18 лет

Срок реализации: 1 год

Протокол № 1 от 28 августа 2024 г.

Приказ № 21 от 28 августа 2024 г.

Учитель информатики

А. Ю. Вихарев

Нолинск, 2024

## ОГЛАВЛЕНИЕ

Пояснительная записка .....	2
Условия реализации программы.....	4
Виды оценки результативности учебных занятий.....	4
Ожидаемые результаты.....	5
Учебный план.....	7
Учебно-тематический план .....	7
Содержание программы.....	9
Календарный учебный график .....	14
Список литературы.....	14

## ОГЛАВЛЕНИЕ

### Пояснительная записка

Современное общество находится в стадии цифровой трансформации, где информационные технологии и электроника проникают во все сферы жизни. От образования и здравоохранения до финансов и развлечений — всё становится зависимым от цифровых инструментов. В этом контексте владение языком программирования становится не просто желательным, а необходимым навыком, позволяющим не только понимать мир вокруг, но и активно участвовать в его создании.

Однако, выбор языка программирования для начинающих — задача непростая. Сегодня существует множество языков, каждый из которых обладает своими преимуществами и особенностями.

Python выходит на первый план, когда речь идет о программировании для школьников. Этот язык завоевал огромную популярность в образовательном сегменте благодаря своей простоте и доступности.

Простота освоения — одна из ключевых особенностей Python. Он не требует глубоких математических знаний и сложных концепций. Синтаксис языка ясен и интуитивен, что делает его идеальным для начинающих программистов.

Наглядность и понятность позволяют быстро получить первые результаты. Простые задачи решаются буквально за несколько строк кода. Это не только мотивирует начинающих, но и позволяет быстро получить практический опыт.

Обширный набор инструментов — еще одна причина, почему Python идеально подходит для первых шагов в программировании.

Стандартные библиотеки Python включают в себя множество готовых функций для работы с текстом, файлами, сетями, графикой и многого другого. Это значительно упрощает написание кода, так как программисту не нужно изобретать велосипед и тратить время на решение стандартных задач.

Множество сторонних библиотек и фреймворков расширяют возможности Python до бесконечности. Независимо от того, какую область программирования выберет в дальнейшем начинающий программист — от веб-разработки до машинного обучения — он найдёт все необходимые инструменты в Python.

Помимо простоты Python обладает еще рядом важных преимуществ:

- Python — современный и востребованный язык, который широко используется в самых разных сферах. Это обеспечивает высокий спрос на специалистов с знанием Python.

- Python поддерживает большое сообщество разработчиков. В сети существует огромное количество ресурсов для обучения, документации и решения проблем. Это делает Python очень дружелюбным языком для новичков, так как всегда есть возможность получить помощь от других программистов.

- Python работает на всех популярных платформах, от Windows до Linux и macOS. Это позволяет программисту работать с Python на любом устройстве.

В итоге Python представляет собой не только простой и доступный язык программирования, но и мощный инструмент для решения реальных задач. Он помогает развить логическое мышление, научиться решать проблемы, получить востребованные навыки и открыть для себя увлекательный мир программирования.

### **Направленность программы**

Дополнительная общеобразовательная программа «Программирование на Python» имеет техническую направленность, что позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление.

### **Цель и задачи**

Целью программы является развитие алгоритмического мышления учащихся посредством изучения программирования на языке Python.

Познавательные задачи:

- начальное освоение языка программирования Python;
- освоение методик работы с базами данных;

- создание приложений с графическим интерфейсом;
- освоение структуры чат-ботов;
- освоение структуры создания игр;

Регулятивные задачи:

- формирование навыков планирования, определения последовательности промежуточных целей с учётом конечного результата;
- освоение способов контроля в форме сопоставления способа действия и его результата с заданным образцом с целью обнаружения отличий от эталона.

Коммуникативные задачи:

- формирование умения работать над проектом в команде;
- овладением умением эффективно распределять обязанности.

**Возраст обучающихся:** 15–18 лет.

## **Условия реализации программы**

Программа рассчитана на 68 академических часа.

Срок реализации — 1 год.

Занятия проводятся 1 раз в неделю по 2 часа.

Для успешной реализации программы необходимо:

- учебная аудитория с 14 рабочими местами для обучающихся;
- моноблочное интерактивное устройство — 1 шт.;
- доска школьная меловая – 1 шт.;
- МФУ (принтер, сканер, копир) – 1 шт.;
- ноутбук для обучающихся – 14 шт.;
- ноутбук для педагога – 1 шт.

## **Виды оценки результативности учебных занятий**

Предметом диагностики и контроля являются учебные проекты обучающихся, а также их личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам программы.

Основой для оценивания деятельности обучающихся являются результаты анализа их учебной деятельности. Оценке подлежит в первую очередь уровень достижения обучающимся минимально необходимых результатов.

Для оценки результативности процесса обучения предусматриваются следующие виды контроля:

- входной (для выяснения знаний, умений и навыков обучающихся на начало учебного года);
- промежуточный (в середине учебного года по пройденным разделам (темам));

итоговый (после завершения всей учебной программы).

Проверка достигаемых обучающимися образовательных результатов проводится в следующих формах:

- текущая диагностика и оценка педагогом деятельности обучающихся;
- текущий контроль, осуществляемый по результатам выполнения практических заданий, мини-проектов, тестов по теме;
- взаимооценка обучающимися работ друг друга или работ, выполненных в группах;
- публичная защита выполненных обучающимися творческих работ (индивидуальных и групповых).

## **Ожидаемые результаты**

Личностные результаты:

- формирование умения самостоятельной деятельности;
- формирование умения работать в команде;
- формирование коммуникативных навыков;
- формирование навыков анализа и самоанализа;
- формирование целеустремленности и усидчивости в процессе творческой, исследовательской работы и учебной деятельности;
- формирование умения распределять/планировать время;
- развитие навыков самопрезентации.

Метапредметные результаты:

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения; владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками;
- работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

## Предметные результаты

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
  - приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
  - 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 конкретного исполнителя;
  - формирование знаний об алгоритмических конструкциях, логических значениях и операциях;
  - овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
  - владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных;
  - владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ;
  - владение элементарными навыками формализации прикладной задачи и документирования программ;
  - формирование умения работать с библиотеками программ;
  - получение опыта использования компьютерных средств представления и анализа данных.
- Учащиеся должны уметь:
- записывать алгоритмические структуры на языке программирования Python;
  - использовать Python для решения заданий единого государственного экзамена;
  - строить алгоритмы методом последовательного уточнения (сверху вниз);
  - использовать основные алгоритмические приемы при решении математических задач;
  - создавать простые игры;

- анализировать текст чужих программ, находить в них неточности,
- оптимизировать алгоритм, создавать собственные варианты решения.

### Учебный план

	Наименование раздела	Количество часов		
		Всего	Теория	Практика
1	Введение	2	1	1
2	Первые программы	12	4	8
3	Линейные алгоритмы	4	1	3
4	Алгоритмы с ветвлением	10	4	6
5	Зачётная работа	2		2
6	Циклические алгоритмы	10	4	6
7	Вспомогательные алгоритмы	4	1	3
8	Строки	4	1	3
9	Массивы	8	2	6
10	Файлы	4	1	3
11	Графика	6	2	4
12	Зачётная работа (защита проекта)	2		2
	<b>Всего</b>	<b>68</b>	<b>21</b>	<b>47</b>

### Учебно-тематический план

	Наименование раздела	Количество часов		
		Всего	Теория	Практика
<b>1</b>	<b>Введение</b>	<b>2</b>	<b>1</b>	<b>1</b>
1.1	Структура курса обучения. Техника безопасности			
1.2	Алгоритмы и исполнители. Этапы решения задач			
<b>2</b>	<b>Первые программы</b>	<b>12</b>	<b>4</b>	<b>8</b>
2.1	Компоненты и интерфейс среды программирования. Структура кода	1	1	
2.2	Вывод данных	1		1
2.3	Оператор присваивания. Выражения и стандартные функции	1	0,5	0,5
2.4	Ввод данных	1	0,5	0,5
2.5	Типы данных	2	0,5	1,5
2.6	Использование и преобразование типов	2	0,5	1,5
2.7	Форматный вывод	2	1	1
2.8	Практическая работа «Запись вы-	2		2

2.5	Тестирование	2	1	3
2.6	История языка и перспективы его развития	2	0,5	1,5
2.7	разжений на языке программирования»	2	1	3
2.8		2	1	3
<b>3</b>	<b>Линейные алгоритмы</b>	<b>4</b>	<b>1</b>	<b>3</b>
3.1	Понятие, представление и примеры линейных алгоритмов	2	1	1
3.2	Практическая работа «Программирование линейных алгоритмов»	2		2
4	Алгоритмы с ветвлением	10	4	6
4.1	Понятие, представление и примеры разветвляющихся алгоритмов	1	1	
4.2	Простые и сложные условия	1		1
4.3	Программирование разветвляющихся алгоритмов полной формы	2	1	1
4.4	Программирование разветвляющихся алгоритмов сокращённой формы.	2	1	1
4.5	Программирование разветвляющихся алгоритмов формы выбора	2	1	1
4.6	Практическая работа «Программирование алгоритмов с ветвлением»	2		2
5	<b>Зачётная работа</b>	<b>2</b>		<b>2</b>
<b>6</b>	<b>Циклические алгоритмы</b>	<b>10</b>	<b>4</b>	<b>6</b>
6.1	Понятие, представление и примеры циклических алгоритмов	1	1	
6.2	Операторы для описания циклов	1		1
6.3	Программирование циклов с условием	2	1	1
6.4	Программирование циклов с параметром	2	1	1
6.5	Программирование вложенных циклов	2	1	1
6.6	Практическая работа «Программирование циклических алгоритмов»	2		2
<b>7</b>	<b>Вспомогательные алгоритмы</b>	<b>4</b>	<b>1</b>	<b>3</b>
7.1	Понятие, представление и примеры вспомогательных алгоритмов	1	1	
7.2	Процедуры и функции	1		1
7.3	Практическая работа «Программирование вспомогательных алгоритмов»	2		2
<b>8</b>	<b>Строки</b>	<b>4</b>	<b>1</b>	<b>3</b>
8.1	Понятие, описание, средства об-	1	1	

7.3	Массивы	2		2
	работки строк			
8.2	Операции со строками	1		1
8.3	Практическая работа «Работа со строками»	2		2
<b>9</b>	<b>Массивы</b>	<b>8</b>	<b>2</b>	<b>6</b>
9.1	Понятие, описание, способы заполнения массивов	2	1	1
9.2	Программирование одномерных массивов	1		1
9.3	Программирование двумерных массивов	1		1
9.4	Сортировка массивов	2	1	1
9.5	Практическая работа «Программирование процессов обработки массивов»	2		2
<b>10</b>	<b>Файлы</b>	<b>4</b>	<b>1</b>	<b>3</b>
10.1	Понятие и средства для работы с файлами	1	1	
10.2	Операции с текстовыми файлами	1		1
10.3	Практическая работа «Работа с текстовыми файлами»	2		2
<b>11</b>	<b>Графика</b>	<b>6</b>	<b>2</b>	<b>4</b>
11.1	Система координат графического экрана	2	1	1
11.2	Графические возможности	2	1	1
11.3	Практическая работа «Создание рисунков»			
12	Зачётная работа (защита проекта)	2		2
	Всего	68	21	47

## Содержание программы

### Введение (2 часа).

1. Введение. Техника безопасности.

Теория: Правила внутреннего распорядка. Техника безопасности в компьютерном классе.

Практическая работа: Регистрация обучающихся в Интернет-сервисах для интерактивного взаимодействия.

2. Алгоритмы и исполнители. Этапы решения задач.

Теория: Понятие алгоритма. Исполнители алгоритмов. Система команд исполнителя. Формы представления алгоритмов: словесное, графическое, алгоритмический язык. Примеры алгоритмов из жизни и учебной деятельности.

Назначение блоков блок-схем при графической форме представления алгоритмов. Этапы решения задач с помощью компьютера: моделирование, алгоритмизация, программирование, тестирование, анализ, сопровождение.

Практическая работа: Исполнители алгоритмов. Система команд исполнителя. Формальное исполнение алгоритмов. Команды меню для управления исполнителем.

### **Первые программы (12 часов).**

1. Компоненты и интерфейс среды программирования. Структура программы.

Теория: Основные понятия языка программирования: оператор, операнды, среда программирования, идентификатор, операции, выражения, алфавит, служебные слова, комментарии. Составляющие среды программирования. Понятие величины. Виды величин: переменные, константы. Структура программы.

#### **2. Вывод данных.**

Практическая работа: Составление и отладка программ с использованием функции вывода данных.

#### **3. Оператор присваивания. Выражения и стандартные функции.**

Теория: Понятие оператора присваивания, взаимосвязь переменной и ее значения. Выражения: арифметические, логические, отношения и другие. Обозначение операций. Приоритеты выполнения операций в выражениях. Правила записи арифметических выражений и стандартных функций. Составление программ с использованием сложных арифметических выражений и стандартных функций. Свойства переменных при использовании оператора присваивания. Обмен значений переменных.

Практическая работа: Составление и отладка программ с использованием оператора присваивания, арифметических выражений, использованию стандартных функций.

#### **4. Ввод-вывод данных.**

Теория: Функция ввода данных. Функция вывода данных: вывод данных, результатов выполнения выражений. Составление диалога.

Практическая работа: Составление и отладка программ с использованием функций ввода-вывода.

#### **5. Типы данных.**

Теория: Типы данных: идентификаторы, размер памяти, диапазон значений, разновидности. Простые типы данных: целые числа, вещественные числа, логические значения. Составные типы данных: множества, списки. Особенности использования.

Практика: Составление и отладка программ на использование различных типов данных.

#### 6. Использование и преобразование типов.

Теория: Преобразование типов: возможности, функции.

Практическая работа: Составление и отладка программ на использование различных типов данных с преобразованием.

#### 7. Форматный вывод.

Теория: Понятие и особенности форматного вывода данных.

Практика: Составление и отладка программ на использование форматного вывода данных.

8. Практическая работа «Запись выражений на языке программирования». Выполнение заданий на составление арифметических выражений и использованию стандартных функций.

### **Линейные алгоритмы (4 часа).**

#### 1. Понятие, представление и примеры линейных алгоритмов.

Теория: Понятие линейного алгоритма. Линейная алгоритмическая структура «Следование», словесная форма представления, представление в виде блок-схемы. Принцип выполнения.

Практическая работа: Программирование линейных алгоритмов.

#### 2. Практическая работа «Программирование линейных алгоритмов».

### **Разветвляющиеся алгоритмы (10 часов).**

#### 1. Понятие, представление и примеры разветвляющихся алгоритмов.

Теория: Понятие разветвляющегося алгоритма. Полная и сокращенная структура «Ветвление», форма выбора, словесная форма представления, представление в виде блок-схем различных форм ветвления. Операторы для описания ветвлений.

#### 2. Простые и сложные условия.

Практическая работа: Применение простых и сложных условий при использовании операторов различных форм ветвления.

#### 3. Программирование разветвляющихся алгоритмов полной формы.

Теория: Использование структуры «Ветвление» для решения задач на основе использования условного оператора полной формы.

Практическая работа: Составление, отладка и выполнение программ для реализации разветвляющегося процесса полной формы.

4. Программирование разветвляющихся алгоритмов сокращенной формы.

Теория: Использование структуры «Ветвление» для решения задач на основе использования условного оператора сокращенной формы.

Практическая работа: Составление, отладка и выполнение программ для реализации разветвляющегося процесса сокращенной формы.

#### 5. Программирование разветвляющихся алгоритмов формы выбора.

Теория: Использование структуры «Ветвление» для решения задач на основе использования оператора выбора.

Практическая работа: Составление, отладка и выполнение программ для реализации разветвляющегося процесса на основе использования формы выбора.

6. Практическая работа «Программирование разветвляющихся алгоритмов».

### **Защита проекта (2 часа).**

### **Циклические алгоритмы (10 часов).**

#### 1. Понятие, представление и примеры циклических алгоритмов.

Теория: Понятие циклических алгоритмов. Тело цикла, параметр цикла, начальное и конечное значение параметра, шаг цикла. Циклы с условием, циклы с параметром, вложенные циклы, словесная форма представления, представление в виде блок-схем различных форм циклов.

#### 2. Операторы для описания циклов.

Практическая работа: Операторы цикла с условием, цикла с параметром (счетчика). Прерывание работы цикла, прерывание итерации цикла.

#### 3. Программирование циклов с условием.

Теория: Циклы с условием, особенности, принцип выполнения.

Практическая работа: Составление, отладка и выполнение программ для реализации циклического процесса с условием.

#### 4. Программирование циклов с параметром.

Теория: Цикл с параметром, особенности, принцип выполнения.

Практическая работа: Составление, отладка и выполнение программ для реализации циклического процесса с параметром.

#### 5. Программирование вложенных циклов.

Теория: Вложенные циклы, особенности, принцип выполнения.

Практическая работа: Составление, отладка и выполнение программ для реализации циклического процесса с использованием вложенных циклов.

#### 6. Практическая работа «Программирование циклических алгоритмов».

### **Вспомогательные алгоритмы (4 часа).**

#### 1. Понятие, представление и примеры вспомогательных алгоритмов.

Теория: Понятие вспомогательного алгоритма. Представление вспомогательного алгоритма в виде блок-схемы. Примеры использования вспомогательных алгоритмов. Описание вспомогательного алгоритма на языке про-

граммирования, формальные и фактические параметры. Локальные и глобальные переменные.

## 2. Процедуры и функции.

Теория: Понятие процедур и функций, передача параметров. Механизм использования вспомогательных алгоритмов. Рекурсия.

Практическая работа: Составление, отладка и выполнение программ, содержащих процедуры и функции.

3. Практическая работа «Программирование вспомогательных алгоритмов».

## Строки (4 часа).

### 1. Понятие, описание, средства обработки строк.

Теория: Строковый тип данных. Функции и методы для работы со строками. Операции над символами и строками.

### 2. Операции над строками.

Практическая работа: Составление программ с использованием строкового типа данных для выполнения обработки строк.

### 3. Лабораторная работа «Работа со строками».

## Массивы (8 часов).

### 1. Понятие, описание, способы заполнения массивов.

Теория: Понятие массива и их применение. Виды массивов: одномерные, двумерные массивы. Идентификатор, индекс, элемент массива.

Описание массивов. Способы заполнения и обработки массивов.

Практическая работа: Способы описания и заполнения одномерных массивов.

### 2. Программирование одномерных массивов.

Практическая работа: Составление, отладка и выполнение программ для реализации процессов по обработке одномерных массивов.

### 3. Программирование двумерных массивов.

Практическая работа: Составление, отладка и выполнение программ для реализации процессов по обработке двумерных массивов.

### 4. Сортировка массивов.

Теория: Понятие и виды сортировок в массивах, принцип реализации.

Практическая работа: Составление, отладка и выполнение программ для реализации сортировки массивов.

5. Практическая работа «Программирование процессов обработки массивов».

## Файлы (4 часа).

### 1. Понятие и средства для работы с файлами.

Теория: Понятие файла. Типы файлов. Этапы работы с файлами. Функции для работы с текстовыми файлами.

2. Операции с текстовыми файлами.

Практическая работа: Составление и отладка программ с использованием текстовых файлов.

3. Практическая работа «Работа с текстовыми файлами».

### **Графика (6 часов).**

1. Система координат графического экрана.

Теория: Графические модули. Система координат графического экрана, текущий указатель. Процедуры для задания цвета, толщины и стиля линии, цвета закрашки.

Практическая работа

2. Графические возможности.

Теория: Графические примитивы, работа с графическими примитивами. Установка цветов. Приемы анимации.

Практическая работа: Составление программ для построения графических изображений на основе использования графических примитивов, иллюстрирующих графические возможности языка Python.

3. Практическая работа «Создание графических рисунков».

### **Защита проекта (2 часа)**

## **Календарный учебный график**

Группа	Учебные недели	Объём учебных часов	Режим работы
№ 1	34	68	суббота, 10.00-11.30 (2 академ. часа)
№ 2	34	68	суббота, 12.00-13.30 (2 академ. часа)

## **Список литературы**

### **Для педагога:**

1. Васильев. А. Программирование на Python в примерах и задачах. — Москва : Эксмо, 2021. — 616 с.

2. Гутман . Г. Библиотека Tkinter: графика, геометрия и логические игры на Питоне : - @Гутман Г.Н., 2021. - 259 с.

3. Дауни. А. Основы Python. Научитесь думать как программист ; пер. с англ. С. Черникова ; [науч. ред. А. Родионов]. — Москва : Манн, Иванов и Фербер, 2021. — 304 с.

4. МакГрат. М. Программирование на Python для начинающих : [перевод с англ. М.А. Райтмана]. – Москва : Эксмо, 2015. — 192 с.

англ. С. Чарльзис; [пер. с англ. Д. С. Чарльзис] — М.: Мир, 1988. — 304 с.

4. Мейер, Р. Программирование. Python. С++. Часть 1 : учебное пособие. — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.

**Для обучающихся:**

1. Поляков, К. Программирование. Python. С++. Часть 1 : учебное пособие. — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.:

3. Поляков, К. Программирование. Python. С++. Часть 1 : учебное пособие. — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.:

**Для обучающихся:**

1. Поляков, К. Программирование. Python. С++. Часть 1 : учебное пособие. — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.: