

**Муниципальное общеобразовательное учреждение  
«Средняя общеобразовательная школа № 6 села Архангельского  
Буденновского района»**

СОГЛАСОВАНО

руководитель  
Центра цифрового  
и гуманитарного образования  
«Точка роста» МОУ СОШ №6  
с. Архангельского

 Ж. А. Першина  
«28» августа 2023г

УТВЕРЖДЕНО

приказом МОУ СОШ №6  
Архангельского  
от «30» августа 2023г. № 354-од  
Директор МОУ СОШ №6  
с. Архангельского



Н.П. Иванова



**ДОПОЛНИТЕЛЬНАЯ  
ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
технической направленности  
«Основы программирования на Python»**

Уровень программы: ознакомительный  
Возрастная категория: 14-17 лет  
Состав группы: до 15 учащихся  
Срок реализации: 1 год  
ID-номер программы в Навигаторе: 24887

Автор составитель:  
Писковатская Галина Николаевна  
учитель информатики  
ЦО «Точка Роста»

Принята на заседании  
педагогического совета  
«30» августа 2023 г.  
протокол №1

с. Архангельское  
2023 год

Дополнительная общеразвивающая программа «Основы программирования на Python» имеет техническую направленность, направление «Информационные технологии», предназначена для организации по нескольким взаимосвязанным направлениям развития личности, таким как общеинтеллектуальное, общекультурное, социальное. Программа ориентирована на углубление знаний и умений по курсу информатики основной школы. Основы инженерного и технического образования – а именно такие специалисты сегодня, да и в ближайшем будущем будут остро нужны стране – закладываются именно в школе. Данная программа ориентирована на один год обучения.

**Программа реализуется на основании нормативно-правовых документов:**

Федеральным законом Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;

Распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р.»Об утверждении Концепции дополнительного образования детей до 2030 года»;

Постановлением Главного государственного санитарного врача Российской Федерации от 28 сентября 2020г. «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания, обучения, отдыха и оздоровления детей и молодежи»;

Приказом Министерства просвещения РФ «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» 27 июля 2022 г. № 629;

Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки Российской Федерации 18 ноября 2015 г. № 09-3242).

Примерными требованиями к содержанию и оформлению образовательных программ дополнительного образования детей (Письмо Департамента молодежной политики, воспитания и социальной поддержки детей Минобрнауки России от 11 декабря 2006 года № 06 - 1844);

**Актуальность программы**

Python –это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. Как считают многие, один из самых используемых языков программирования в мире. Программа направлена на развитие логического и пространственного мышления слушателя, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

**Цель программы**

Создание условий для достижения обучающимися результатов развития в личностном, предметном, метапредметном направлениях, обеспечивающих их социальную адаптацию в области программирования, ИКТ на профессиональном уровне.

Развить личность обучающегося, способного к творческому самовыражению через овладение основами программирования на одном из самых востребованных языков программирования Python.

Понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации; научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

**Задачи программы**

### ***Образовательные:***

Познакомить с возможностями и особенностями современного языка программирования Python;

Сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;

Изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами); научить применять функции при написании программ на языке программирования Python;

научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Сформировать систему знаний, умений и навыков, необходимых в работе программиста;

Обучить технологии работы со справочниками по языку программирования и поиску нужной информации на сайтах разработчиков программного обеспечения;

Обучить приёмам предъявления результатов проделанной работы на конференциях и конкурсах;

Обучить коммуникативным навыкам при совместной работе над проектом.

### ***Развивающие:***

Развить познавательные потребности и способности школьников

Развивать творческие способности, алгоритмическое и логическое мышление;

Развивать представления о возможностях применения современных компьютерных технологий в профессиональной деятельности;

Развивать внимательность и наблюдательность, прививать навыки аккуратности и точности в работе;

Повысить интеллектуальный уровень и расширить интеллектуальные навыки; □ Создать мотивацию к постоянному самообразованию.

### ***Воспитательные:***

Воспитывать навыки взаимодействия при командной работе над проектом;

Воспитывать чувство ответственности за результаты деятельности;

Способствовать формированию культуры программирования;

Способствовать формированию у воспитанников культуры создания и этики представления проектов на конференциях и в Интернете;

Сформировать осознанное отношение к выбору будущей профессии.

***Предметные:*** изучить принципы структурного, функционального и объектного программирования на примере языка программирования Python. Научиться формулировать и анализировать алгоритмы, составлять и отлаживать программы.

***Метапредметные:*** научиться моделировать различные процессы реального мира в компьютере и использовать такое моделирование как инструмент познания.

***Личностные:*** гармоничное развитие творческих способностей и логического мышления учащихся.

## **1. Планируемые результаты**

**Основным результатом обучения** является формирование вектора развития обучающихся с упором на формирование у них алгоритмического мышления.

### ***Личностные результаты:***

сформированность мировоззрения, соответствующего современному уровню развития науки и техники;

осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов;

сформированность представлений о мире профессий, связанных с программированием, и требованиях, предъявляемых различными востребованными профессиями, такими как программист, системный администратор;

навыки сотрудничества в образовательной, учебно-исследовательской, проектной и других видах деятельности;

навыки взаимо- и самооценки, навыки рефлексии.

***Метапредметные результаты:***

владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;

способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания.

***Предметные результаты:***

навыки алгоритмического мышления и понимание необходимости формального описания алгоритмов;

владение стандартными приёмами написания программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ на языке Python;

знание особенностей структуры программы, представленной на языке Python,

представление о модулях, входящих в состав среды Python,

возможности и ограничения использования готовых модулей,

представление о величине, ее характеристиках,

знание что такое операция, операнд и их характеристики,

знание принципиальные отличия величин, структурированных и не структурированных,

представление о таких структурах данных, как число, текст, кортеж, список, словарь,

представление о составе арифметического выражения;

знание математических функций, входящих в Python, представление о логических выражениях и входящих в них операндах, операциях и функциях,

умение записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить,

знание основных операторов языка Python, их синтаксис,

представление о процессе исполнения каждого из операторов,

умение разрабатывать программы обработки числовой и символьной информации,

умение разрабатывать программы (линейные, разветвляющиеся и с циклами),

представление о значении полноценных процедур и функций для структурно-ориентированного языка высокого уровня,

правила описания функций в Python и построение вызова,

принципиальные отличия между формальными, локальными и глобальными переменными.

## **Содержание тем курса**

### **10 класс. (72 часа)**

#### **1 . Основы охраны труда. Организация рабочего места. (2 ч.)**

Валеологические аспекты и требования безопасности при работе за компьютером. Безопасность в Интернете.

Обучающиеся знакомятся с правилами поведения в компьютерном классе, правилами охраны труда, рекомендациями по организации рабочего места.

Практическая работа 1.1. Организация рабочего места.

***Обучающиеся должны уметь:***

правильно организовать рабочее место;

следить за временными нормами работы за персональным компьютером;

использовать компьютер, не нанося вреда своему здоровью; Обучающиеся знакомятся с правилами безопасного поведения при работе в сети Интернет.

## **2. Знакомство с языком Python (4 ч.)**

Общие сведения о языке Python, преимущества языка, сферы применения. Установка программы Python на компьютер. Режимы работы Python. Использование командной строки интерпретатора. Выбор редактора. Работа со средой программирования. Сохранение, открытие и запуск программ. Получение помощи. Знакомство с языком Python и написание простейших программ. Структура программы на языке Python Комментарии.

Практическая работа 2.1. Установка программы Python.

Практическая работа 2.2. Режимы работы с Python.

**Обучающиеся должны знать / понимать:**

понятие программы; структура программы на Python; режимы работы с Python.

**Обучающиеся должны уметь:**

выполнить установку программы;

выполнить простейшую программу в интерактивной среде;

написать комментарии в программе.

## **3. Переменные и выражения. Операции. Организация ввода и вывода данных. (8 ч.)**

Литеральные константы, числа, строки. Использование кавычек. Объединение строковых констант. Метод format/

Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Объекты. Примеры использования переменных и констант. Логические и физические строки. Отступы. Операторы и их применение. Выражения. Операции. Порядок выполнения операций. Краткая запись математических операций и порядок вычисления. Изменение порядка вычисления. Математические функции. Композиция. Ассоциативность. Элементарные действия с числами.

Ввод и вывод данных. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных.

Практическая работа 3.1. Работа со справочной системой.

Практическая работа 3.2. Переменные.

Практическая работа 3.3. Выражения.

Практическая работа 3.5. Задачи на элементарные действия с числами.

**Обучающиеся должны знать / понимать:**

общую структуру программы;

типы данных;

целые, вещественные типы данных и операции над ними;

оператор присваивания.

## **4. Программирование ветвящихся алгоритмов. (10 ч.)**

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Поток команд. Ветвящиеся алгоритмы. Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python. Решение задач на программирование ветвящихся алгоритмов.

Практическая работа 4.1. Логические выражения.

Практическая работа 4.2. "Условный оператор".

Практическая работа 4.3. Множественное ветвление.

Практическая работа 4.4. "Условные операторы".

**Обучающиеся должны знать / понимать:**

назначение условного оператора;

способ записи условного оператора;

логический тип данных;

логические операторы or, and, not.

**Обучающиеся должны уметь:**

использовать условный оператор;  
создавать сложные условия с помощью логических операторов.

**5. Программирование циклических алгоритмов. (10 ч.)**

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Реализация циклических алгоритмов. Решение задач с использованием циклов. Операторы управления циклом. Оператор break. Оператор continue.

Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 5.1. Решение задачи с циклом while.

Практическая работа 5.2. Решение задачи с циклом for.

Практическая работа 5.3. Реализация циклических алгоритмов.

Практическая работа 5.4. Случайные числа.

Практическая работа 5.5. Решение задач с циклом.

**Обучающиеся должны знать / понимать:**

циклы с условием и их виды;

правила записи циклов с условием;

назначение и особенности использования цикла с параметром;

формат записи цикла с параметром;

примеры использования циклов различных типов.

**Обучающиеся должны уметь:**

определять вид цикла, наиболее удобный для решения поставленной задачи;

использовать цикл с условием;

определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

**6. Функции (12ч.)**

Создание функций. Параметры функций. Локальные и глобальные переменные. Зарезервированные слова «global», «nonlocal». Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Значения аргументов по умолчанию. Ключевые аргументы. Переменное число параметров. Ключевые параметры. Оператор «return».

Строки документации. Аннотации. Решение задач с использованием функций. Рекурсивные функции.

Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 6.1. Создание функций.

Практическая работа 6.2. Локальные переменные.

Практическая работа 6.3. Решение задач с использованием функций.

Практическая работа 6.4. Рекурсивные функции.

**Обучающиеся должны знать / понимать:**

понятие функции;

способы описания функции;

принципы структурного программирования;

понятие локальных переменных подпрограмм;

понятие формальных и фактических параметров подпрограмм;

способ передачи параметров.

**Обучающиеся должны уметь:**

создавать и использовать функции;

использовать механизм параметров для передачи значений.

**7. Строки (8 ч.)**

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 7.1. Строки

Практическая работа 7.2. Решение задач со строками.

**Учащиеся должны знать / понимать:**

назначение строкового типа данных; операторы для работы со строками; процедуры и функции для работы со строками; операции со строками.

**Учащиеся должны уметь:**

описывать строки; соединять строки; находить длину строки ;вырезать часть строки; находить подстроку в строке; находить количество слов в строке.

## **8. Модули (4 ч.)**

Для чего нужны модули. Оформление модулей. Импорт модуля. Оператор from ... import .

Имя модуля, name. Создание собственных модулей. Функция dir. Пакеты.

Практическая работа 8.1. Использование модулей.

Практическая работа 8.2. Создание модулей.

**Обучающиеся должны знать / понимать:**

для чего нужны модули.

**Обучающиеся должны уметь:**

использовать готовые модули;  
создавать собственные модули.

## **9. Структуры данных. (12 ч.)**

Списки. Объекты и классы. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков.

Срезы списков. Удаление списка. Клонирование списков. Списочные параметры.

Функция range. Списки: примеры решения задач. Матрицы. Вложенные списки. Матрицы.

Строки и списки.

Генераторы списков в Python.

Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Последовательности.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств.

Операции, допустимые над множествами: объединение, пересечение, разность, включение.

Оператор определения принадлежности элемента множеству.

Практическая работа 9.1. Списки.

Практическая работа 9.2. Решение задач со списками.

**Обучающиеся должны знать / понимать:**

сложные типы данных;

назначение строкового типа данных;

операторы для работы со строками;

процедуры и функции для работы со строками;

операции со строками;

способ описания списка;

способ доступа к элементам списка;

способ описания кортежа;

способ описания словаря;

операции, выполняемые со списками, кортежами и словарями;

понятие множества;

способы описания множества;

операторы работы с множествами.

**Обучающиеся должны уметь:**

описывать строки;  
 соединять строки;  
 находить длину строки;  
 вырезать часть строки;  
 находить подстроку в строке;  
 находить количество слов в строке;  
 описывать списки;  
 вводить элементы списка;  
 выводить элементы списка;  
 выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;  
 использовать вложенные списки;  
 приводить примеры использования вложенных списков (матриц);  
 описывать множества;  
 определять принадлежность элемента множеству;  
 вводить элементы множества;  
 выводить элементы множества.

**10. Стиль программирования и отладка программ. (2 ч.)**

Процесс разработки программного обеспечения. Стиль программирования. Отладка программ.

**Обучающиеся должны знать / понимать:**

что такое стиль программирования;  
 правила именования объектов;  
 основные рекомендации при написании программ.

**Обучающиеся должны уметь:**

определять вид ошибок и находить ошибки в программе.  
 выполнять тестирование и отладку программ.

**Календарно-тематическое планирование**

№ п/п	Название темы	Количество часов
1.	Основы охраны труда. Организация рабочего места.	2
2.	Знакомство с языком Python.	4
3.	Переменные и выражения. Организация ввода и вывода данных. Операции.	8
4.	Программирование ветвящихся алгоритмов.	10
5.	Программирование циклических алгоритмов	10
6.	Функции	12
7.	Строки	8
8.	Модули	4
9.	Структуры данных	12
10.	Стиль программирования и отладка программ	2
Всего:		72 часа

**Поурочное планирование**

Дата	Кор. даты	№ п/п	Название раздела, темы	Кол-во часов
		<b>1</b>	<b>Основы охраны труда. Организация рабочего места.</b>	<b>2</b>
05.09		1.	Основы охраны труда	1
05.09		2.	Организация рабочего места.	1
		<b>2</b>	<b>Знакомство с языком Python.</b>	<b>4</b>
12.09		3.	Общие сведения о языке Python.	1
12.09		4.	Установка программы Python	1
19.09		5.	Что такое программа. Структура программ на языке Python.	1
19.09		6.	Режимы работы с Python	1
		<b>3</b>	<b>Переменные и выражения. Операции. Организация ввода и вывода данных.</b>	<b>8</b>
26.09		7.	Переменные	1
26.09		8.	Выражения.	1
03.10		9.	Операции.	1
03.10		10.	Порядок выполнения операций	1
10.10		11.	Работа со справочной системой	1
10.10		12.	Элементарные действия с числами	1
17.10		13.	Ввод данных с клавиатуры.	1
17.10		14.	Пример скрипта, использующего ввод и вывод данных. Вывод данных на экран	1
		<b>4</b>	<b>Программирование ветвящихся алгоритмов.</b>	<b>10</b>
24.10		15.	Логические выражения	1
24.10		16.	Логический оператор or	1
31.10		17.	Логические оператор and	1
31.10		18.	Логические оператор not	1
07.11		19.	Сложные условные выражения	
07.11		20.	Условный оператор if	1
14.11		21.	Множественное ветвление	1
14.11		22.	Способ записи условного оператора	1
21.11		23.	Логический тип данных	1
21.11		24.	Реализация ветвления в языке Python	1
		<b>5</b>	<b>Программирование циклических алгоритмов</b>	<b>10</b>
28.11		25.	Понятие цикла. Тело цикла	1
28.11		26.	Числа Фибоначчи	1
05.12		27.	Оператор цикла с условием while	1
05.12		28.	Оператор цикла с параметром for	1
12.12		29.	Случайные числа	
12.12		30.	Бесконечные циклы	1
19.12		31.	Вложенные циклы	1
19.12		32.	Функция randrange	1
26.12		33.	Функция random	1
26.12		34.	Реализация циклических алгоритмов	1
		<b>6</b>	<b>Функции</b>	<b>12</b>
09.01		35.	Создание функций	1
09.01		36.	Параметры функций	1
16.01		37.	Локальные переменные	1
16.01		38.	Глобальные переменные	1
23.01		39.	Зарезервированные слова «global», «nonlocal»	1
23.01		40.	Поток выполнения	1
30.01		41.	Функции, возвращающие результат.	1

30.01	42.	Анонимные функции, инструкция lambda	1
06.02	43.	Значения аргументов по умолчанию	1
06.02	44.	Переменное число параметров. Оператор «return».	1
13.02	45.	Рекурсивные функции. Вычисление факториала.	1
13.02	46.	Числа Фибоначчи.	1
	<b>7</b>	<b>Строки</b>	<b>8</b>
20.02	47.	Составной тип данных - строка	1
20.02	48.	Доступ по индексу	1
27.02	49.	Длина строки и отрицательные индексы	1
27.02	50.	Преобразование типов.	1
05.03	51.	Применение цикла для обхода строки.	1
05.03	52.	Оператор in. Модуль string	1
12.03	53.	Срезы строк. Сравнение строк	1
12.03	54.	Операторы для всех типов последовательностей	1
	<b>8</b>	<b>Модули</b>	<b>4</b>
19.03	55.	Имя модуля, name. Создание собственных модулей	1
19.03	56.	Функция dir. Пакеты.	1
26.03	57.	Импорт модуля. Оператор from ... import .	1
26.03	58.	Оформление собственных модулей.	1
	<b>9</b>	<b>Структуры данных</b>	<b>12</b>
02.04	59.	Строки	1
02.04	60.	Срезы строк	1
09.04	61.	Списки	1
09.04	62.	Срезы списков	1
16.04	63.	Генераторы списков в Python.	1
16.04	64.	Кортежи	1
23.04	65.	Словари	1
23.04	66.	Последовательности	1
07.05	67.	Матрицы	1
07.05	68.	Множества	1
14.05	69.	Оператор определения принадлежности элемента множеству.	1
14.05	70.	Ссылки	1
	<b>10</b>	<b>Стиль программирования и отладка программ</b>	<b>2</b>
21.05	71.	Стиль программирования	1
21.05	72.	Отладка программ	1
		<b>Итого</b>	<b>72</b>