

Муниципальное общеобразовательное учреждение Белоярского района  
«Средняя общеобразовательная школа № 2 г. Белоярский»

СОГЛАСОВАНО  
заместитель директора по ВР



Истомина О.Ю.



директор  
Исаченко Н.В.

Исаченко Н.В.

Приказ от 12.01.2026 г. № 150

**Дополнительная общеобразовательная  
общеразвивающая  
программа технической направленности  
«ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON»**

*для обучающихся 8-11 классов*

*срок реализации программы – 1 год*

Автор программы:  
педагог дополнительного образования, учитель  
информатики  
Карауловская Ольга Евгеньевна

г. Белоярский, 2026

## Пояснительная записка

Программа разработана в соответствии с нормативно-правовыми документами в соответствии с ФЗ-273 «Об образовании в РФ», Концепцией развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р, приказом Министерства Просвещения от 27 июля 2022 г. N 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

Программа представляет собой дополнительную общеразвивающую программу технической направленности.

Программа интегрирована с проектами в области технологического образования: Национальной технологической олимпиадой (НТО), ее профилем «Информационная безопасность». Программа ориентирована на развитие инженерного мышления, навыков командной работы и проектной деятельности, необходимых для участия в инженерных соревнованиях.

Содержание обучения, представленное в программе «Язык программирования Python», позволяет вести обучение школьников в режиме актуального познания.

Активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения, информационной безопасностью.

Программа служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся.

*Отличительной особенностью* программы является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования.

*Уровень усвоения* – базовый.

*Целевая аудитория:* обучающиеся 8-11 классов (занятия могут проводиться в разновозрастных группах по 15 человек)

*Срок освоения программы* - 1 год.

*Объём программы* – 72 часа.

*Цель программы* - овладение базовыми понятиями теории алгоритмов; освоение понятия алгоритма и особенности реализации алгоритмов в виде программ, написанных на языке программирования Python, подготовка к участию в олимпиаде НТО и Всероссийской олимпиады школьников по профилю «Информационная безопасность».

*Задачи программы:*

- познакомиться с понятиями «алгоритм», «язык программирования»;
- научиться составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python;
- научиться работать с графическими средствами языка программирования Python;
- научиться отлаживать и тестировать программы, делать выводы о работе этих программ.

*Потенциальные партнеры по сотрудничеству в реализации программы:* автономное учреждение дополнительного образования Ханты-Мансийского автономного округа – Югры «Мастерская талантов «Сибиряк» <https://xn--b1acg7bcla5a1cv.xn--p1ai/> ; кружковое движение НТИ <https://kruzhok.org/> ; флагманский проект «Движения первых» – конкурс «Большая перемена».

### **Планируемые результаты курса**

В рамках программы «Язык программирования Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.
- приобретение опыта участия в проектной и соревновательной деятельности.

## Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Язык программирования Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Педагогическая ценность контроля заключается в том, что он даёт всестороннюю информацию о способностях учащихся к анализу или синтезу, оценочным суждениям и позволяет оценить эффективность учебного труда для каждого из них.

Диагностика и контроль — необходимые части учебного процесса, но увеличение их доли неизбежно приводит к сокращению времени на изучение материала. Поэтому столь важно извлечение максимума информации об учащихся за минимальное время. Контроль и диагностика должны быть действенными. Поэтому необходимо анализировать результаты проверки и принимать меры по коррекции образовательного процесса. От этого зависит, станут ли способы оценивания уровня достижений учащихся результативными.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

Созданными внешними образовательными продуктами учащиеся могут пополнять собственные портфолио.

Проверка достигаемых школьниками результатов производится в следующих формах:

- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка учителем деятельности школьников в виде трёх контрольных работ по следующим темам: «Алгоритмизация. Знакомство с Python»; «Основные алгоритмические конструкции»; «Структурированные типы данных».

Итоговый контроль проводится в конце курса. Он организуется в форме подготовки к участию в отборочном этапе олимпиады НТО и выполнении тестирующих заданий.

*Аппаратное обеспечение:*

1. IBM PC-совместимый компьютер.

2. Процессор не ниже Pentium-100.
3. Оперативная память не меньше 64 Мб.

*Программное обеспечение:*

4. Операционная система: Windows XP (или выше).
5. Одна из сред разработки

#### **Учебно-методическое обеспечение**

1. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
2. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
3. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
4. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с. – 84 с.

## Тематический план курса

Наименование разделов и тем	Количество часов	
	Всего	Прак. занятия
<b>Модуль 1</b>	<b>36</b>	
<b>Раздел 1. Основы алгоритмизации</b>	<b>7</b>	<b>7</b>
<b>Раздел 2. Знакомство с Python</b>	<b>14</b>	<b>14</b>
2.1. Структура программы на языке Python	2	2
2.2. Числовые типы данных	4	4
2.3. Подпрограммы	2	2
2.4. Использование графического модуля	4	4
Контрольная работа № 1	2	2
<b>Раздел 3. Основные алгоритмические конструкции</b>	<b>15</b>	<b>15</b>
3.1. Циклы	6	6
3.2. Условный оператор. Оператор выбора	6	6
3.3. Средства отладки программ	3	3
<b>Модуль 2</b>	<b>36</b>	
<b>Раздел 3. Основные алгоритмические конструкции</b>	<b>8</b>	<b>8</b>
3.4. Константы	2	2
3.5. Компьютерная анимация	4	4
Контрольная работа № 2	2	2
<b>Раздел 4. Структурированные типы данных</b>	<b>22</b>	<b>22</b>
4.1. Массивы	6	6
4.2. Типизированные константы	2	2
4.3. Строковый тип данных	4	4
4.4. Записи	2	2
4.5. Файловый тип данных	6	6
<b>Раздел 5. Соревновательная деятельность</b>	<b>8</b>	<b>8</b>
5.1 НТО: профиль «Информационная безопасность». Знакомство с регламентом, разбор задач прошлых лет. Подготовка к участию в отборочном этапе. Разбор заданий инженерного и предметного туров	4	4
5.2 Командообразование. Разбор индивидуальных и командных заданий II тура	4	4
<b>ВСЕГО</b>	<b>72</b>	<b>72</b>

№	По плану	Тема
---	----------	------

1.	<b>Январь, неделя 1</b>	Основы алгоритмизации
2.		Основы алгоритмизации
3.	<b>Январь, неделя 2</b>	Основы алгоритмизации
4.		Основы алгоритмизации
5.	<b>Январь, неделя 3</b>	Основы алгоритмизации
6.		Основы алгоритмизации
7.	<b>Январь, неделя 4</b>	Основы алгоритмизации
8.		Структура программы на языке Python
9.	<b>Февраль, неделя 1</b>	Структура программы на языке Python
10.		Числовые типы данных
11.	<b>Февраль, неделя 2</b>	Числовые типы данных
12.		Числовые типы данных
13.	<b>Февраль, неделя 3</b>	Числовые типы данных
14.		Подпрограммы
15.	<b>Февраль, неделя 4</b>	Подпрограммы
16.		Использование графического модуля
17.	<b>Март, неделя 1</b>	Использование графического модуля
18.		Использование графического модуля
19.	<b>Март, неделя 2</b>	Использование графического модуля
20.		Контрольная работа № 1
21.	<b>Март, неделя 3</b>	Контрольная работа № 1
22.		Циклы
23.	<b>Март, неделя 4</b>	Циклы
24.		Циклы
25.	<b>Апрель, неделя 1</b>	Циклы
26.		Циклы
27.	<b>Апрель, неделя 2</b>	Циклы
28.		Условный оператор. Оператор выбора
29.	<b>Апрель, неделя 3</b>	Условный оператор. Оператор выбора
30.		Условный оператор. Оператор выбора
31.	<b>Апрель, неделя 4</b>	Условный оператор. Оператор выбора
32.		Условный оператор. Оператор выбора
33.	<b>Май, неделя 1</b>	Условный оператор. Оператор выбора
34.		Средства отладки программ
35.	<b>Май, неделя 2</b>	Средства отладки программ
36.		Средства отладки программ
37.	<b>Май, неделя 3</b>	Средства отладки программ
38.		Константы
39.	<b>Май, неделя 4</b>	Константы

40.		Компьютерная анимация
41.	<b>Сентябрь, неделя 1</b>	Компьютерная анимация
42.		Компьютерная анимация
43.		Компьютерная анимация
44.		Контрольная работа № 2
45.	<b>Сентябрь, неделя 2</b>	Контрольная работа № 2
46.		Массивы
47.	<b>Сентябрь, неделя 3</b>	Массивы
48.		Массивы
49.	<b>Сентябрь, неделя 4</b>	Массивы
50.		Массивы
51.	<b>Октябрь, неделя 1</b>	Массивы
52.		Типизированные константы
53.	<b>Октябрь, неделя 2</b>	Типизированные константы
54.		Строковый тип данных
55.	<b>Октябрь, неделя 3</b>	Строковый тип данных
56.		Строковый тип данных
57.	<b>Октябрь, неделя 4</b>	Строковый тип данных
58.		Строковый тип данных
59.	<b>Ноябрь, неделя 1</b>	Записи
60.		Записи
61.	<b>Ноябрь, неделя 2</b>	Файловый тип данных
62.		Файловый тип данных
63.	<b>Ноябрь, неделя 3</b>	Файловый тип данных
64.		Файловый тип данных
65.	<b>Ноябрь, неделя 4</b>	Файловый тип данных
66.		Файловый тип данных
67.	<b>Декабрь, неделя 2</b>	Контрольная работа № 3
68.		Контрольная работа № 3
69.	<b>Декабрь, неделя 3</b>	Итоговый зачет
70.		Итоговый зачет
71.	<b>Декабрь, неделя 4</b>	Резерв
72.		Резерв

# Содержание курса

## Раздел 1. Основы алгоритмизации

*Учащиеся должны знать / понимать:*

- понятие алгоритма;
- понятие исполнителя;
- назначение и основные команды среды исполнителя;
- типы алгоритмов;
- свойства алгоритма;
- язык блок-схем.

*Учащиеся должны уметь:*

- составлять несложные алгоритмы для исполнителя;
- записывать алгоритм разными способами;
- определять исполнителя алгоритма.

Алгоритмы. Способы записи алгоритма. Исполнители алгоритмов. Типы алгоритмов: вспомогательные, циклические, разветвляющиеся. Определение и свойства алгоритма.

## Раздел 2. Знакомство с Python

### Тема 2.1. Структура программы на языке Python

*Учащиеся должны знать / понимать:*

- назначение и основные команды среды разработки;
- общую структуру программы;
- назначение и виды оператора вывода.

*Учащиеся должны уметь:*

- пользоваться интерфейсом среды программирования Python
- использовать команды редактора;
- составлять и запускать программы;
- организовывать вывод данных.

Язык программирования Python и его характерные особенности. Структура программы на языке Python. Простейшая программа. Среда разработки. Элементы языка Python. Создание и исполнение программ в среде разработки. Операторы вывода.

### Тема 2.2. Числовые типы данных

*Учащиеся должны знать / понимать:*

- понятие типа данных;
- целые, вещественные типы данных и операции над ними;
- понятие переменной;
- оператор присваивания;
- назначение и виды оператора ввода.

*Учащиеся должны уметь:*

- определять тип числовых данных;
- объявлять необходимые переменные;
- записывать арифметические выражения.

Переменные. Типы данных в языке Python. Простые типы данных. Целые и вещественные типы. Значения. Оператор присваивания. Операции, допустимые с переменными и значениями целого и вещественного типа.

### **Тема 2.3. Подпрограммы**

*Учащиеся должны знать / понимать:*

- назначение подпрограмм;
- отличия процедур и функций;
- понятие формальных и фактических параметров.

*Учащиеся должны уметь:*

- объявлять процедуры и функции в программе Python;
- вызывать подпрограммы из основной программы.

Подпрограмма. Процедуры и функции. Параметры, формальные и фактические параметры.

### **Тема 2.4. Использование графического модуля**

*Учащиеся должны знать / понимать:*

- понятие модуля;
- назначение и возможности графического модуля.

*Учащиеся должны уметь:*

- подключить графический модуль;
- инициализировать графический режим;
- использовать графические примитивы.

Модули. Графический модуль, назначение и возможности. Графический экран (режим). Основные графические примитивы. Управление цветом. Штриховка.

## **Контрольная работа № 1.**

### **Раздел 3. Основные алгоритмические конструкции**

#### **Тема 3.1. Циклы**

*Учащиеся должны знать / понимать:*

- понятие и назначение цикла;
- цикл со счётчиком;
- циклы с условием;
- понятие генератора случайных чисел;
- понятие символьного типа;

- назначение и возможности модуля **CRT**;
- понятие кода клавиши, расширенного кода клавиши.

*Учащиеся должны уметь:*

- использовать все виды циклов для повторения блока действий в программе;
- определять оптимальный вид оператора цикла для решения поставленной задачи;
- использовать генератор случайных чисел;
- использовать символьные переменные и константы;
- принимать коды и расширенные коды клавиш: символьных и служебных.

Цикл. Цикл со счётчиком. Цикл с предусловием и цикл с постусловием. Генератор случайных чисел. Символьный тип данных

### **Тема 3.2. Условный оператор. Оператор выбора**

*Учащиеся должны знать / понимать:*

- понятие и назначение условного оператора;
- назначение оператора выбора;
- алгоритм поиска максимального/минимального элемента;
- логические выражения;
- способы тестирования программ.

*Учащиеся должны уметь:*

- использовать условный оператор, оператор выбора при составлении программ;
- осуществлять выбор типа условного оператора/оператора выбора для оптимального решения поставленной задачи;
- составлять сложные логические выражения;
- использовать алгоритм поиска максимального/минимального элемента последовательности;
- составлять тестовую таблицу, тестировать готовую программу.

Условный оператор. Полная и неполная формы условного оператора. Оператор выбора.

Алгоритм поиска максимального / минимального элемента последовательности.

Тестирование готовой программы.

### **Тема 3.3. Средства отладки программ**

*Учащиеся должны знать / понимать:*

- механизм отладки;
- возможности отслеживания значений переменных;
- способы пошагового выполнения программы;
- метод дихотомии.

*Учащиеся должны уметь:*

- вывести в окно отладки имена переменных;

- произвести пошаговое выполнение программы;
- протестировать программу, выявлять и исправлять ошибки;
- находить корни произвольных уравнений методом дихотомии.

Отладка. Окно пользователя. Пошаговое выполнение программы.

Алгоритм обмена значений двух переменных. Поиск корней уравнения методом дихотомии.

### **Тема 3.4. Константы**

*Учащиеся должны знать / понимать:*

- понятие константы;
- принципы преобразования экранных координат.

*Учащиеся должны уметь:*

- объявить константу, использовать её значение в программе;
- строить график произвольной функции в заданном масштабе и расположении.

Константы. Объявление константы. Использование константы в программе.

Преобразования экранных координат. Построение графика функции.

### **Тема 3.5. Компьютерная анимация**

*Учащиеся должны знать / понимать:*

- принципы компьютерной анимации;

*Учащиеся должны уметь:*

- реализовать несложное движение объектов на экране, в том числе и с переменной формой.

Компьютерная анимация. Передвижение объекта по заданной траектории. Передвижение объекта с изменяющейся формой.

Операции инкремента и декремента, их преимущества перед оператором присваивания.

Организация задержки в программе.

## **Контрольная работа № 2.**

### **Раздел 4. Структурированные типы данных**

#### **Тема 4.1. Массивы**

*Учащиеся должны знать / понимать:*

- понятие массива;
- понятие двумерного массива как массива массивов;
- способы поиска максимального/минимального элемента массива;
- способы сортировки — сортировка выбором и пузырьковая.

*Учащиеся должны уметь:*

- объявлять одномерные и двумерные массивы;
- использовать массивы для хранения данных в программе;

- осуществлять поиск максимального/минимального элемента в одномерном массиве;
- произвести сортировку одномерного массива одним из двух способов: сортировкой выбором или пузырьковой сортировкой.

Массив. Одномерный массив. Двумерный массив. Объявление массивов. Обработка массивов: поиск элемента по заданным признакам, заполнение массива, вывод массива на экран. Сортировка массива. Сортировка выбором. Пузырьковая сортировка.

#### **Тема 4.2. Типизированные константы**

*Учащиеся должны знать / понимать:*

- понятие типизированной константы;
- область применения типизированных констант;
- отличие типизированных констант от переменных и от констант;
- принципы передвижения рисованного объекта по экрану без следа.

*Учащиеся должны уметь:*

- объявлять типизированные константы;
- использовать массив-буфер для сохранения области экрана при передвижении объекта по экрану.

Типизированные константы.

Алгоритм передвижения объекта по экрану без оставления следа. Использование битового образа для хранения образа объекта. Использование массива-буфера для хранения области экрана.

#### **Тема 4.3. Строковый тип данных**

*Учащиеся должны знать / понимать:*

- строковый тип данных;
- понятие строк как массива символов;
- допустимые действия над строковыми данными.

*Учащиеся должны уметь:*

- объявлять и использовать в программе величины строкового типа;
- использовать стандартные процедуры и функции обработки строк.

Строковый тип данных. Строки как массив символов. Стандартные строковые процедуры и функции.

#### **Тема 4.4. Записи**

*Учащиеся должны знать / понимать:*

- понятие типа «запись»;

- понятия «поле», «запись»;
- область применения типа «запись»;
- понятие пользовательского типа.

*Учащиеся должны уметь:*

- создавать пользовательский тип;
- использовать записи для хранения базы данных.

Пользовательские типы данных. Поля записи. Поле-массив. Поле-запись. Массив записей.

Заполнение массива записей. Вывод значения записи на экран.

#### **Тема 4.5. Файловый тип данных**

*Учащиеся должны знать / понимать:*

- понятие файлового типа;
- отличия и область применения типизированных и текстовых файловых типов;
- порядок работы с данными файлового типа.

*Учащиеся должны уметь:*

- определить оптимальный тип файловых данных для решения конкретной задачи;
- ассоциировать файловую переменную с файлом на диске;
- открыть файл для чтения или записи;
- записать/прочитать информацию из файла.

Файловый тип данных. Типизированные файлы. Текстовые файлы.

#### **Тема 5.1 Подготовка к участию в отборочном (I) этапе НТО**

*Учащиеся должны уметь:*

- Писать простые программы и читать программные коды, в том числе неизвестных авторов.
- Пользоваться нейросетями для упрощения работы.
- Проводить анализ скомпилированного программного кода.
- Проводить анализ вредоносного воздействия и определить его природу и источник.
- Искать скрытую информацию в незнакомом объекте.
- Находить и закрывать уязвимости в веб-сервисах.
- Выполнять задания профиля, связанные со следующими инженерными компетенциями: криптография, веб-безопасность, анализ инцидентов