

Муниципальное бюджетное общеобразовательное учреждение  
«Средняя общеобразовательная школа № 58» г. Брянска.

***Аннотация к рабочей программе***  
учебного курса внеурочной деятельности  
«Язык программирования Python»

Рабочая программа учебного предмета «Физика. Базовый уровень» обязательной предметной области «Естественнонаучные предметы» разработана в соответствии с пунктом 18.2.2 ФГОС СОО и реализуется 1 год в 10 классе.

Рабочая программа разработана учителем в соответствии с положением о рабочих программах и определяет организацию образовательной деятельности учителя в школе по определенному *учебному курсу внеурочной деятельности.*

Рабочая программа *учебного предмета* является частью ООП СОО определяющей:

- содержание;
- планируемые результаты (личностные, метапредметные и предметные);
- тематическое планирование с учетом рабочей программы воспитания и возможностью использования ЭОР/ЦОР.

Рабочая программа обсуждена и принята решением методического объединения и согласована заместителем директора по учебно-воспитательной работе МБОУ СОШ №58.

Дата: 29.08.2023 г.

Муниципальное бюджетное общеобразовательное учреждение  
«Средняя общеобразовательная школа № 58 г. Брянска»

Выписка

из основной образовательной программы среднего общего образования

Рассмотрено  
методическое объединение  
учителей математики и информатики  
протокол № 1 от 30.08.03

Согласовано  
заместитель директора по ВР  
Зайцева А.Г.  
30.08.03

Рабочая программа  
учебного курса внеурочной деятельности  
«Язык программирования Python»  
для уровня среднего общего образования  
Срок реализации: 1 год (10класс)

Составитель:  
учитель информатики  
Качаев С.Н.

Выписка от 30.08.2023  
Директор И.В. Волобуева



## Пояснительная записка

Данный курс «Язык программирования Python» во внеурочной деятельности и рассчитан на 34 часа.

Изменение взглядов на предмет информатики как науки, её место в системе научного знания требует существенных изменений в содержании образования по информатике. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей среды.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей, связанных с познанием, если они видят смысл в сотрудничестве с одноклассниками и учителем.

Содержание обучения, представленное в программе курс «Языки программирования. Python», позволяет вести обучение школьников в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов — блок-схем, алгоритмов, программ — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики.

Проявления трудолюбия, целеустремленности, возникающие при воплощении замыслов учащихся в рамках курс «Языки программирования. Python», стимулируют развитие индивидуально-личностных качеств школьников.

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 дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

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

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

В рамках предлагаемого курса «Языки программирования. Python» изучение основ программирования на языке Python — это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Исключительно велика роль

программирования для формирования мышления школьников, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления учащихся.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения по отношению к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции — одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных школьников. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

#### **Цели изучения курса:**

понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;

овладеть базовыми понятиями теории алгоритмов;

научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

#### **Задачи курса:**

познакомить с понятиями алгоритма, вычислимой функции, языка программирования;

научить составлять и читать блок-схемы;

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.

#### **Методы обучения**

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать

лично-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала учителем, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе — обучение учащихся навыкам самостоятельной творческой деятельности.

### **Формы организации учебных занятий**

Организация учебного процесса предусматривает дистанционную формы деятельности, когда учащийся вне уроков самостоятельно выполняет на компьютере практические задания.

### **Планируемые результаты курса**

В рамках курса «Языки программирования. Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умением составлять алгоритмы для решения задач;
- умением реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владением основными навыками программирования на языке Python;
- умением отлаживать и тестировать программы, написанные на языке Python.

### **Способы оценивания уровня достижений учащихся**

Предметом диагностики и контроля в курсе «Основы программирования на языке 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.

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.).

*Проверка достигаемых учащимися результатов производится в следующих формах:*

- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка учителем деятельности школьников;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальной образовательной программой освоения курса;

**Итоговый контроль** проводится в конце всего курса в виде тестирования.

**Программное обеспечение:**

Операционная система: Windows 7 (или выше).

Среда разработки: Python 3.3 (или выше),

Учебно-тематический план

## ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№ урока	Тема урока	Кол-во часов	Дата проведения
	<b>Тема 1. Переменные и выражения</b>		
1.	Знакомство с языком Python	1	
2.	Практическая работа 1.1. Работа со справочной системой	1	
3.	Практическая работа 1.2. Переменные	1	
4.	Практическая работа 1.3. Выражения	1	
5.	Практическая работа 1.4. Ввод и вывод данных	1	
6.	Практическая работа 1.5. Задачи на элементарные действия с числами	1	
	<b>Тема 2. Условные предложения</b>		
7.	Практическая работа 2.1. Логические выражения	1	
8.	Практическая работа 2.2. Условный оператор «if»	1	
9.	Практическая работа 2.3. Множественное ветвление «if-else»	1	
10.	Практическая работа 2.4. "Условные операторы"	1	
11.	Зачетная работа № 1. "Составление программ с ветвлением".	1	
	<b>Тема 3. Циклы</b>		
12.	Практическая работа 3.1. "Числа Фибоначчи"	1	
13.	Практическая работа 3.2. Решение задачи с циклом «for»	1	

14.	Практическая работа 3.3. Реализация циклических алгоритмов	1	
15.	Практическая работа 3.4. Случайные числа	1	
16.	Практическая работа 3.5. Решение задач с циклом	1	
17.	Творческая работа № 1. "Циклы"	1	
	<b>Тема 4. Функции</b>		
18.	Практическая работа 4.1. Создание функций	1	
19.	Практическая работа 4.2. Локальные переменные	1	
20.	Практическая работа 4.2. Локальные переменные	1	
21.	Практическая работа 4.3. Решение задач с использованием функций	1	
22.	Практическая работа 4.4. Рекурсивные функции	1	
23.	Практическая работа 4.4. Рекурсивные функции	1	
	<b>Тема 5. Строки - последовательности символов</b>		
24.	Практическая работа 5.1. Строки	1	
25.	Примеры решения задач со строками	1	
26.	Практическая работа 5.2. Решение задач со строками.	1	
	<b>Тема 6. Сложные типы данных</b>		
27.	Практическая работа 6.1. Списки	1	
28.	Практическая работа 6.2. Решение задач со списками	1	
	<b>Тема 7. Стиль программирования и отладка программ</b>		
29.	Матрицы	1	
30.	Кортежи	1	

31.	Введение в словари	1	
32.	Множества в языке Python	1	
33.	Стиль программирования	1	
34.	Отладка программ	1	

### Содержание тем учебного курса

#### Тема 1. Переменные и выражения

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

Выражения. Операции. Порядок выполнения операций. Математические функции.

Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 1.1. Работа со справочной системой, Практическая работа 1.2. Переменные, Практическая работа 1.3. Выражения, Практическая работа 1.4. Ввод и вывод данных. Практическая работа 1.5. Задачи на элементарные действия с числами.

Учащиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- операторы ввода-вывода.

Учащиеся должны уметь:

- пользоваться интерфейсом среды программирования Python;
- использовать команды редактора;
- организовывать ввод и вывод данных;
- записывать арифметические выражения.

#### Тема 2. Условные предложения

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.



Практическая работа 2.1. Логические выражения Практическая работа 2.2. "Условный оператор" Практическая работа 2.3. Множественное ветвление Практическая работа 2.4. "Условные операторы"

Учащиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not; Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

### Тема 3. Циклы

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. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 3.1. "Числа Фибоначчи" Практическая работа 3.2. Решение задачи с циклом for. Практическая работа 3.3. Реализация циклических алгоритмов Практическая работа 3.4. Случайные числа Практическая работа 3.5. Решение задач с циклом.

Учащиеся должны знать / понимать:

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся 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;

### Тема 4. Функции

Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция `lambda`. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 4.1. Создание функций

Практическая работа 4.2. Решение задач с использованием функций

Практическая работа 4.3. Решение задач с использованием функций

Практическая работа 4.4. Рекурсивные функции

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Тема 5. Строки - последовательности символов

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор `in`. Модуль `string`. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 5.1. Строки

Практическая работа 6.2. 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.

Тема 6. Сложные типы данных

Списки. 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.

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 6.1. Списки.

Практическая работа 6.2. Решение задач со списками.

Учащиеся должны знать / понимать:

- сложные типы данных;
- 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.

Учащиеся должны уметь:

- 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.

Тема 7. Стиль программирования и отладка программ

Стиль программирования. Отладка программ.  
Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать / понимать:

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

Учащиеся должны уметь:

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

### **Литература и источники**

1. Домашняя страница Python [www.python.org](http://www.python.org). Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
4. Сайт проекта Open Book Project [openbookproject.net](http://openbookproject.net) содержит серию практических примеров на Python Криса Мейерса.
5. Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
6. Python. Справочник Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.