



**Муниципальное казённое общеобразовательное учреждение
средняя общеобразовательная школа № 7
город Узловая Тульской области**

Рассмотрено на заседании МО
Руководитель МО
М. В. Белоусова / 
Протокол № 1 от 29.08.2018 г.

Согласовано
Заместитель директора
по УВР
Ю. А. Тихонова / 

Утверждаю
И. о. директора МКОУ СОШ № 7
Е. Н. Шевчук
Приказ № 33 от 31.08.2018 г.
на основании решения
Педагогического совета,
протокол № 1 от 29.08.2018 г.



РАБОЧАЯ ПРОГРАММА
ЭЛЕКТИВНОГО КУРСА ПО ИНФОРМАТИКЕ

«Программирование на языке Python»

8 класс

Разработана: Бражко Е. Л.,
учителем информатики и ИКТ

Пояснительная записка

Элективный курс по информатике «Программирование на языке Python» представляет собой вводный курс по программированию, дающий представление о базовых понятиях структурного программирования (данных, операциях, переменных, ветвлениях в программе, циклах и функциях, массивах) на языке Python.

Выбор Python обусловлен тем, что это язык, обладающий рядом преимуществ перед другими языками для начинающих изучать программирование: ясность кода, быстрота реализации. Python — это язык программирования, который становится всё более популярным в последнее время. Он пригоден для решения разнообразных задач и предлагает те же возможности, что и другие языки программирования: динамичность, поддержку ООП и кроссплатформенность. В заданиях тематического блока «Программирование» контрольно-измерительных материалов ЕГЭ по информатике и ИКТ пятым языком представлен язык программирования Python.

Python позволяет писать очень компактные и удобочитаемые программы. Изучая программирование на Python, учащиеся прочнее усваивают основы алгоритмизации, приобщаются к алгоритмической культуре, познают азы профессии программиста.

Критерии выбора языка программирования:

- Свободная кроссплатформенная реализация
- Удобная среда разработки, адаптированная для учебного процесса
- Широкая распространенность, поддерживаемость и развиваемость языка
- Возможность автоматической проверки
- Понятность языка и лёгкое восприятие учащимися

Достоинства языка Python:

- Современный язык программирования, с богатыми возможностями и большой стандартной библиотекой
- Используется для промышленного программирования, для решения «домашних» прикладных задач и для обучения
- Кроссплатформенная, свободная реализация
- Хорошо документирован (<https://www.python.org/doc/>)
- Свободен для Windows, имеется во всех Linux-дистрибутивах

Особенности языка Python:

- Интерпретируемый
- Блоки кода выделяются величиной отступа:

```
max = A[0]
for elem in A:
    if elem > max:
        max = elem
```
- Динамическая типизация переменных

```
a = 2 + 2
a = "hello"
```
- Полностью объектно-ориентированный

```
print(" ".join(map(str,
sorted(list(map(int, input().split()))))))
```

Python — язык на все вкусы:

- Высокоуровневые структуры данных: списки, множества, ассоциативные массивы
- Элементы функционального программирования

- Разработка GUI
- Сетевые приложения
- Web-приложения
- Базы данных

Предельно лаконичный и понятный синтаксис:

Pascal	Python
<code>function gcd (a, b: integer): integer;</code>	<code>def gcd(a, b):</code>
<code>var t: integer;</code>	<code>while b > 0:</code>
<code>begin</code>	<code> (a, b) = (b, a%b)</code>
<code> while b > 0 do</code>	<code> return a</code>
<code> begin</code>	
<code> t := a mod b;</code>	
<code> a := b;</code>	
<code> b := t</code>	
<code> end;</code>	
<code> gcd := a</code>	
<code>end</code>	

Версии 2 и 3 языка Python:

- сейчас существует две основные версии языка — 2 и 3, которые не полностью совместимы;
- много наработок именно по версии 2;
- версия 3 более совершенна с точки зрения дизайна языка;
- тем не менее, версия 2 более не развивается, поэтому рекомендуется сразу же использовать версию 3.

Концепция курса

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 «Программирование на языке 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 личностно-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.

Планируемые результаты курса

В рамках курса «Программирование на языке 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.

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Программирование 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.

Программное обеспечение:

1. Операционная система: Windows XP/7/8/10 или Linux
2. Язык программирования: Python 3 (версия 3.6.6 или выше)
3. Среда разработки: Wing 101 (Wing Python IDE)

Курс рассчитан на 35 часов (1 час в неделю) и основан на авторском курсе «Основы программирования на Python», разработанном С. В. Шапошниковой, МБОУ ДОД ЦД(Ю)ТТ «Городской» г. Липецка (sshap@yandex.ru) и распространяемом согласно лицензии GNU FDL:

<https://younglinux.info/python.php> (<http://window.edu.ru/resource/825/76825>)

https://younglinux.info/sites/default/files/pdf/python_structured_programming.pdf

Учебно-тематический план

№ п/п	Тема	Кол-во часов
1	Знакомство с языком Python	2
2	Переменные и выражения	4
3	Условные выражения	5
4	Циклы	6
5	Функции	4
6	Строки — последовательности символов	3
7	Сложные типы данных	7
8	Стиль программирования и отладка программ	4
	<i>Итого:</i>	<i>35</i>

Содержание тем учебного курса

Тема 1. Знакомство с языком Python

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1 «Установка программы Python»

Практическая работа 1.2 «Режимы работы с Python»

Тест № 1 «Знакомство с языком Python»

Учащиеся должны знать/понимать:

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

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Тема 2. 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 2.1 «Работа со справочной системой»

Практическая работа 2.2 «Переменные»

Практическая работа 2.3 «Выражения»

Практическая работа 2.4 «Ввод данных с клавиатуры. Вывод данных на экран»

Практическая работа 2.5 «Задачи на элементарные действия с числами»

Тест № 2 «Выражения и операции»

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 3. Условные выражения

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 3.1 «Логические выражения»

Практическая работа 3.2 «Условный оператор»

Практическая работа 3.3 «Множественное ветвление»

Практическая работа 3.4 «Условные операторы»

Самостоятельная работа № 1 «Решение задач по теме “Условные операторы”»

Зачетная работа № 1 «Составление программ с ветвлением»

Тест № 3 «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 4. 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 4.1 «Числа Фибоначчи»

Практическая работа 4.2 «Решение задачи с циклом for»

Практическая работа 4.3 «Реализация циклических алгоритмов»

Практическая работа 4.4 «Случайные числа»

Практическая работа 4.5 «Решение задач с циклом»

Самостоятельная работа № 2 «Составление программ с циклом»

Тест № 4 «Циклы»

Творческая работа № 1 «Циклы»

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 5. 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 5.1 «Создание функций»

Практическая работа 5.2 «Локальные переменные»

Практическая работа 5.3 «Решение задач с использованием функций»

Практическая работа 5.4 «Рекурсивные функции»

Самостоятельная работа № 3 по теме «Функции»

Тест № 5 «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 6. 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 6.1 «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 7. 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 7.1 «Списки»

Практическая работа 7.2 «Решение задач со списками»

Тест № 7 «Списки»

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 8. 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.

Календарно-тематическое планирование

<i>Номер урока</i>	<i>Содержание учебного материала</i>	<i>Дата урока</i>
1.	Общие сведения о языке. Практическая работа 1.1 «Установка программы Python»	
2.	Режимы работы. Практическая работа 1.2 «Режимы работы с Python»	
3.	Переменные. Практическая работа 2.1 «Работа со справочной системой». Практическая работа 2.2 «Переменные»	
4.	Выражения. Практическая работа 2.3 «Выражения»	
5.	Ввод и вывод. Практическая работа 2.4 «Ввод данных с клавиатуры. Вывод данных на экран»	
6.	Элементарные действия с числами. Практическая работа 2.5 «Задачи на элементарные действия с числами»	
7.	Логические выражения и операторы. Практическая работа 3.1 «Логические выражения»	
8.	Условный оператор. Практическая работа 3.2 «Условный оператор»	
9.	Множественное ветвление. Практическая работа 3.3 «Множественное ветвление»	
10.	Реализация ветвления в языке Python. Практическая работа 3.4 «Условные операторы»	
11.	Самостоятельная работа № 1 «Составление программ с ветвлением»	
12.	Оператор цикла с условием. Практическая работа 4.1 «Числа Фибоначчи»	
13.	Оператор цикла for. Практическая работа 4.2 «Решение задачи с циклом for»	
14.	Вложенные циклы. Практическая работа 4.3 «Реализация циклических алгоритмов»	
15.	Случайные числа. Практическая работа 4.4 «Случайные числа»	
16.	Примеры решения задач с циклом. Практическая работа 4.5 «Решение задач с циклом»	
17.	Самостоятельная работа № 2 «Составление программ с циклом»	
18.	Создание функций. Практическая работа 5.1 «Создание функций»	
19.	Локальные переменные. Практическая работа 5.2 «Локальные переменные»	
20.	Примеры решения задач с использованием функций. Практическая работа 5.3 «Решение задач с использованием функций»	
21.	Рекурсивные функции. Практическая работа 5.4 «Рекурсивные функции»	
22.	Строки	
23.	Срезы строк. Практическая работа 6.1 «Строки»	
24.	Примеры решения задач со строками. Практическая работа 6.2 «Решение задач со строками»	

<i>Номер урока</i>	<i>Содержание учебного материала</i>	<i>Дата урока</i>
25.	Списки	
26.	Срезы списков. Практическая работа 7.1 «Списки»	
27.	Списки: примеры решения задач. Практическая работа 7.2 «Решение задач со списками»	
28.	Матрицы	
29.	Кортежи	
30.	Введение в словари	
31.	Множества в языке Python	
32.	Стиль программирования	
33.	Отладка программ	
34.	Итоги курса «Программирование на языке Python»	
35.	Что дальше?	

Перечень учебно-методического обеспечения

1. Ресурсы сайта С. В. Шапошниковой (<https://younglinux.info>), содержащие авторские программы:
<https://younglinux.info/python.php> (<http://window.edu.ru/resource/825/76825>)
<https://younglinux.info/algorithm>
https://younglinux.info/sites/default/files/pdf/python_structured_programming.pdf
2. Сайт Д. П. Кириенко. Московский институт открытого образования. Школа № 179
<http://server.179.ru/wiki/?page=DenisKirienko/Python>
3. Ресурсы сайта профессора, доктора технических наук, учителя информатики высшей категории, автора учебников по информатике К. Ю. Полякова:
<http://kpolyakov.spb.ru/school/probook/python.htm>
http://kpolyakov.spb.ru/download/ch10-8_python.pdf
<http://kpolyakov.spb.ru/download/slides10-8py.zip>
<http://kpolyakov.spb.ru/download/py8class.zip>
<http://kpolyakov.spb.ru/download/py9class.zip>
4. Сайт «Дистанционная подготовка по информатике» <https://informatics.msk.ru>
Курс «Учим Python» <https://informatics.msk.ru/course/view.php?id=95>
5. Курс информатики на Python 3 http://judge.mipt.ru/mipt_cs_on_python3/
6. Официальный сайт Python <https://www.python.org>
Документация, руководства по Python <https://www.python.org/doc/>
7. Среда разработки Wing 101 <http://www.wingware.com/downloads/wing-101>
Документация, руководства по Wing Python IDE <http://www.wingware.com/doc/intro/tutorial>