

Александр Котенок

Уроки программирования
на
Python



Учебник и самоучитель
для
учащихся и студентов

Рабочая книга
для
учителя и преподавателя

Унеча • 2020

Содержание

Введение	3
Глава 1. Программы линейной структуры	4
Урок 1. Интерактивный режим программирования	4
Урок 2. Сценарный режим программирования	7
Урок 3. Строковый тип данных	12
Урок 4. Целочисленный тип данных	14
Урок 5. Тип данных вещественных чисел	17
Урок 6. Способы подключения модуля	20
Урок 7. Математические функции модуля math	22
Урок 8. Программы линейной структуры	24
Глава 2. Ветвления в программах	
Глава 3. Циклы в программах	
Глава 4. Функции пользователя	
Глава 5. Списки	
Глава 6. Массивы	
Глава 7. Кортежи	
Глава 8. Строки	
Глава 9. Множества	
Глава 10. Словари	
Глава 11. Работа с файлами	
Глава 12. Модуль Tkinter	
Глава 13. Программы с графическим интерфейсом	
Глава 14. Программы в ООП стиле	

Введение

Учащимся и студентам

Эта книга для тех, кто решил заглянуть в интересную «страну программирования». Программирование – это область знаний о создании программ для компьютера. Как наука она имеет достаточно обширную терминологию. Не усвоив теоретического минимума знаний, стать программистом невозможно.

С другой стороны, программирование – это технология создания программ для компьютера. Поэтому, необходима ежедневная работа на компьютере по созданию, пока, простейших консольных программ. Хотя, консольные программы имеют не очень привлекательный вид, по сравнению с приложениями с графическим интерфейсом, освоить азы программирования, можно только на многочисленных примерах этих небольших программ. Другого пути в страну программирования нет. Как невозможно освоить математический анализ без знания арифметики и алгебры, а аналитическую геометрию – без школьной геометрии. Чтобы стать спортсменом необходимы постоянные тренировки. Чтобы стать программистом необходимо регулярно решать задачи на компьютере, составляя программы. В начале задачи будут простые, а далее все усложняться. Программирование – это область знаний и область технологии одновременно. Чтобы стать настоящим программистом, необходимо формировать у себя знания, умения, навыки математика, сценариста, дизайнера. Если вы будете регулярно выполнять все задания в книге, то вам откроется заветная дверь в «страну программирования». Книгу можно использовать как самоучитель, но быстрее и эффективнее вы усвоите материал и выработаете навыки программирования с учителем. Пусть ваш путь познания постоянно сопровождается возгласом – «Ура, заработало!» Успехов, вам, в самом интересном занятии – создании софта для компьютера!

Учителям и преподавателям

Эта книга – не только учебник и самоучитель, но и методическое руководство для учителя. Имея такую книгу, вам не нужно писать конспекты к уроку. Уроки по программированию для вас уже написаны. Вам остается карандашом проставить дату урока, вписать или вложить дополнения к теоретическому материалу. Возможно, вы измените количество, и некоторые листинги и файлы программ. Учитывая дифференцированный подход в обучении и творческие возможности ваших учащихся, изменяйте и дополняйте задания для самостоятельной работы. Успехов, вам, в таком замечательном деле – передаче знаний и навыков программирования молодежи!

Условные обозначения

Дидактической единицей обучения является урок. Поэтому данная книга, как и все самоучители иностранных языков, построены по структуре урока. Каждый урок имеет единую структуру. Для удобства восприятия и экономии места в книге, элементы урока обозначены прописной буквой соответствующего номера урока. В табл. 1.0. приведены эти условные обозначения.

Таблица 1.0. Дидактические элементы урока

Символы	Дидактическая единица урока	Комментарий
Урок	Название урока	Тема урока

К	Конспект	Теоретические сведения по теме
П	Практикум	Выполнение тренировочных упражнений по образцам. Использование готовых файлов для тестирования и модификации кода.
С	Самостоятельная работа	Выполнение самостоятельно заданий по составлению программ.
Р	Резюме	Краткий результат обучения.
Л	Лексикон	Термины программирования. «Чтобы вспомнить, нужно уметь забыть!» (А.К.) Лексический принцип дидактики.

Глава 1. Программы линейной структуры

Урок 1. Интерактивный режим программирования

Ц 1. Познакомиться с интерактивным режимом работы в среде программирования IDLE Python3. Научиться использовать интерактивный режим работы в качестве калькулятора.

К 1. Первый релиз языка программирования Python появился в 1991 году. Создателем языка является Гвидо ван Россум (Guido van Rossum). Хотя графическим символом языка является змея питон, название языка, создатель дал в честь «Monty Python» – английской комической группы «Монти Пайтон». По-русски название языка можно произносить «Питон» или «Пайтон». Язык Python поставляется вместе со средой разработки программ IDLE (Python GUI). Среда разработки программ – это инструмент, упрощающий создание программ. Среда IDLE Python 3 содержит: текстовый редактор с подсветкой синтаксиса языка, интерпретатор языка, средства запуска программы на выполнение, синтаксический анализатор. Синтаксический анализатор – компонент среды программирования, проверяющий код программы на отсутствие синтаксических ошибок. После запуска программы на выполнение, синтаксический анализатор проверяет строки кода программы на соответствие синтаксису языка Python. Интерпретатор переводит каждую строку кода языка Python – языка высокого уровня, на машинный язык команд процессора. Если ошибок не будет обнаружено ни в одной строке кода, то программа запускается на выполнение. При наличии синтаксической ошибки в строке кода, дальнейшая проверка прекращается. Попытка запуска программы терпит неудачу. Пока не будут исправлены все синтаксические ошибки, программа остается неработоспособной. В табл. 1.1 приведена подсветка синтаксиса кода языка Python.

Таблица. 1.1 Подсветка синтаксиса языка Python

Цвет	Программный элемент кода	Примеры
Красный	Кодировка, комментарии, сообщения об ошибках	<pre># -*- coding: utf-8 -*- # Первая программа SyntaxError: can't assign to literal</pre>

Фиолетовый	Ключевые (зарезервированные) слова языка	<code>a = int(input("a = "))</code> <code>b = float(input("b = "))</code>
Черный	Имена переменных, литералы, операторы чисел	<code>p = 2*(a + b)</code> <code>d = math.sqrt(a*a + b*b)</code>
Зеленый	Строковые константы	<code>s = "Python"</code>
Голубой	Результаты вывода данных	<code>p = 14.0</code>
Оранжевый	Ключевые слова импорта модуля	<code>from math import *</code>
Коричневый	Знак приглашения интерактивного режима	<code>>>></code>

Среда IDLE Python3 работает в двух режимах – интерактивном (режиме калькулятора) и сценарном (программном). После запуска среды программирования включен интерактивный режим, признаком которого является наличие знака `>>>` и курсора. Пользователь вводит строки с инструкциями языка, и после нажатия клавиши Enter, сразу получает результат. Этот режим можно использовать, как мощный калькулятор, для вычисления математических выражений (рис 1.1).

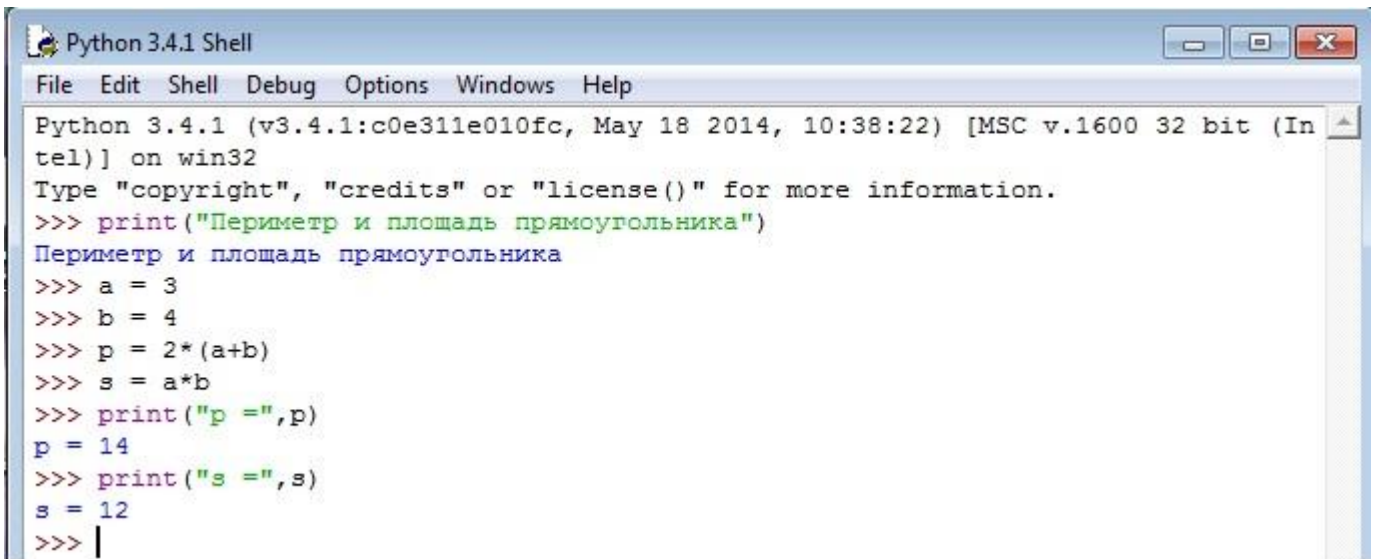
```

*Python 3.4.1 Shell*
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 5+8
13
>>> 7-27
-20
>>> 8*9
72
>>> 2**10
1024
>>> 5.8+15.2
21.0
>>> 45.8*2343.45
107330.009999999998
>>> 82345.67/976.5
84.327363031234
>>> 45.78*(35.8-12.457)**20
1.0563532894695585e+29
>>> |

```

Рис 1.1. Использование интерактивного режима в качестве калькулятора

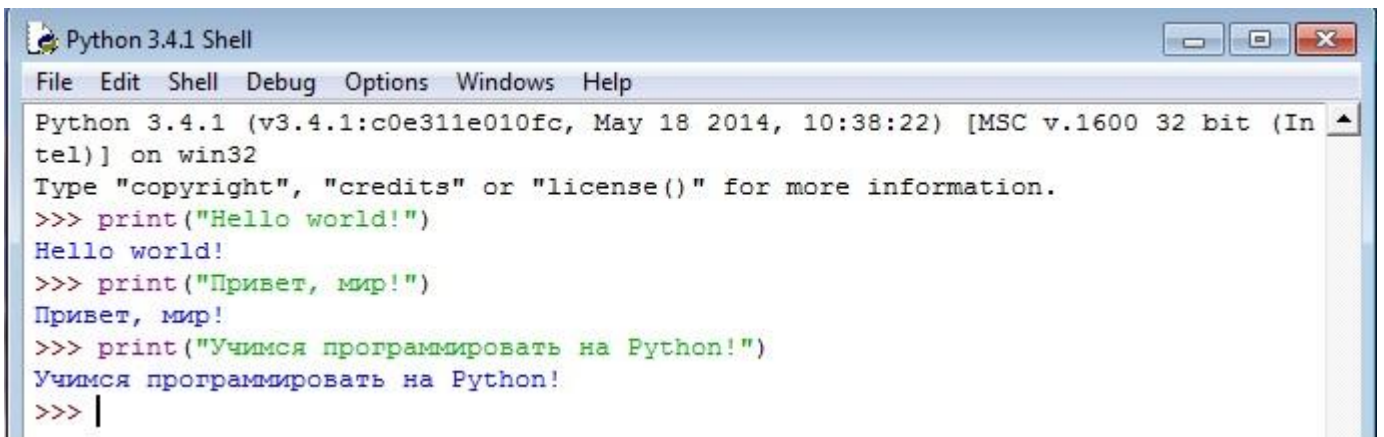
В интерактивном режиме можно проверять работу операторов, встроенных функций и синтаксических конструкций языка, выполнять и небольшие программы, которые используют переменные (рис 1.2).



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Периметр и площадь прямоугольника")
Периметр и площадь прямоугольника
>>> a = 3
>>> b = 4
>>> p = 2*(a+b)
>>> s = a*b
>>> print("p =", p)
p = 14
>>> print("s =", s)
s = 12
>>> |
```

Рис 1.2. Использование переменных в интерактивном режиме

Тот, кто рискнул стать программистом, должен начинать с «первой программы программиста». Эта программа выводит на экран сообщение ("Hello World!"). Не будем нарушать данную традицию. Выполним «первую программу программиста» в интерактивном режиме (рис 1.3).



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello world!")
Hello world!
>>> print("Привет, мир!")
Привет, мир!
>>> print("Учимся программировать на Python!")
Учимся программировать на Python!
>>> |
```

Рис 1.3. Первая программа программиста в интерактивном режиме

П 1. Внимательно прочитайте текст заданий практикума. Откройте IDLE Python3 и выполните их.

- 1.1. В интерактивном режиме выполните математические вычисления с целыми числами.
 $787+392$; $941-537$; $63*59$; $640/40$; $16**2$; $59//4$; $67\%4$;
- 1.2. В интерактивном режиме вычислите степени числа 4 с показателем от 3 до 10.
- 1.3. В интерактивном режиме выполните математические вычисления с десятичными дробями.
 $37.49+72.5$; $16.523-45.16$; $4.5*6.2$; $24.8/0.4$; $25.64**14$;
- 1.4. В интерактивном режиме вычислите математические выражения.
 $(85.4-65.7)/(9.4+8.35)$; $8.5**2+4.24*8.4$; $(75.4-15.4)**2$; $9.3**3+8.45**3$; $43.9-(18.8+5.3)*3/2$;
- 1.5. В интерактивном режиме выполните «первую программу программиста» – наберите `print("Hello World!")` и нажмите клавишу Enter.
- 1.6. В интерактивном режиме выполните «первую программу программиста России» – наберите `print("Привет, мир!")` и нажмите клавишу Enter.

- 1.7. В интерактивном режиме выполните аналог «первой программы программиста» – наберите `print("Изучаем Python")` и нажмите клавишу Enter.
- 1.8. В интерактивном режиме, используя переменные, вычислите площадь треугольника по формуле $s = a*h/2$.

С 1. Внимательно прочитайте текст заданий практикума. Откройте IDLE Python3 и выполните их.

- 1.1. В интерактивном режиме выполните математические вычисления с целыми числами.
 $23+345$; $547-874$; $45*28$; $240/20$; $15**2$; $47//5$; $47\%5$;
- 1.2. В интерактивном режиме вычислите степени числа 2 с показателем от 5 до 15.
- 1.3. В интерактивном режиме выполните математические вычисления с десятичными дробями.
 $56.34+82.7$; $28.173-37.25$; $2.5*2.4$; $12.5/0.5$; $108.24**5$;
- 1.4. В интерактивном режиме вычислите математические выражения.
 $(25.8-15.2)/(5.4+2.31)$; $3.4**2+7.65*4.2$; $(15.8-10.4)**2$; $4.6**3+3.25**3$; $23.5-(3.8+4.9)*7/4$;
- 1.5. В интерактивном режиме выведите сообщение "Учимся программировать на Python".
- 1.6. В интерактивном режиме, используя переменные, вычислите площадь треугольника по основанию a и высоте h .

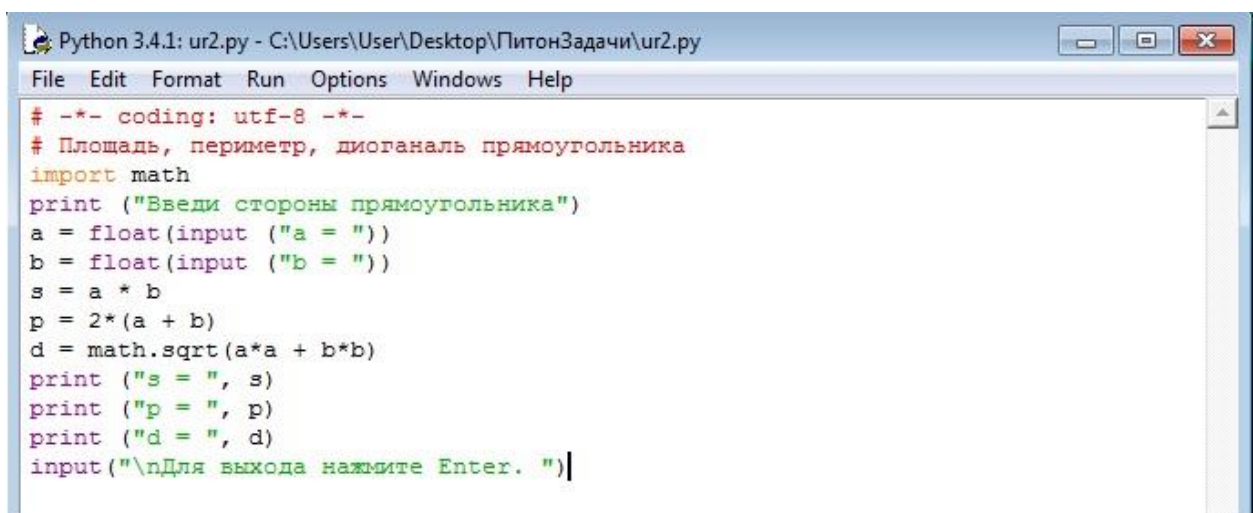
Р 1. Вы умеете работать в интерактивном режиме среды программирования IDLE Python3. Научились использовать интерактивный режим работы в качестве калькулятора.

Л 1. Python • IDLE Python3 • Гвидо ван Россум • Среда программирования • Язык высокого уровня • машинный язык • Интерпретатор • Код программы • Синтаксический анализатор • Подсветка синтаксиса • Интерактивный режим •

Урок 2. Сценарный режим программирования

Ц 2. Познакомиться со сценарным режимом работы в среде программирования IDLE Python3. Уметь создавать «первую программу программиста» и ее аналоги.

К 2. Для входа в сценарный режим необходимо выполнить команду меню: File\New File. Откроется окно текстового редактора, в котором можно набирать код программы или открыть готовый файл. Открыть файл можно командой File\ Open, не выходя из интерактивного режима. Код программы будет помещен в окно текстового редактора IDLE Python3 (рис 2.1).



```
Python 3.4.1: ur2.py - C:\Users\User\Desktop\ПитонЗадачи\ur2.py
File Edit Format Run Options Windows Help
# -*- coding: utf-8 -*-
# Площадь, периметр, диагональ прямоугольника
import math
print ("Введи стороны прямоугольника")
a = float(input ("a = "))
b = float(input ("b = "))
s = a * b
p = 2*(a + b)
d = math.sqrt(a*a + b*b)
print ("s = ", s)
print ("p = ", p)
print ("d = ", d)
input ("\nДля выхода нажмите Enter. ")
```

Рис 2.1. Сценарный режим IDLE Python 3

После открытия файла, сразу сохраните его под новым именем, а потом модифицируйте код программы. Программа на языке Python является файлом с расширением .py или .pyw . Запускается программа на выполнение из среды программирования командой Run\Run Module (рис 2.2).

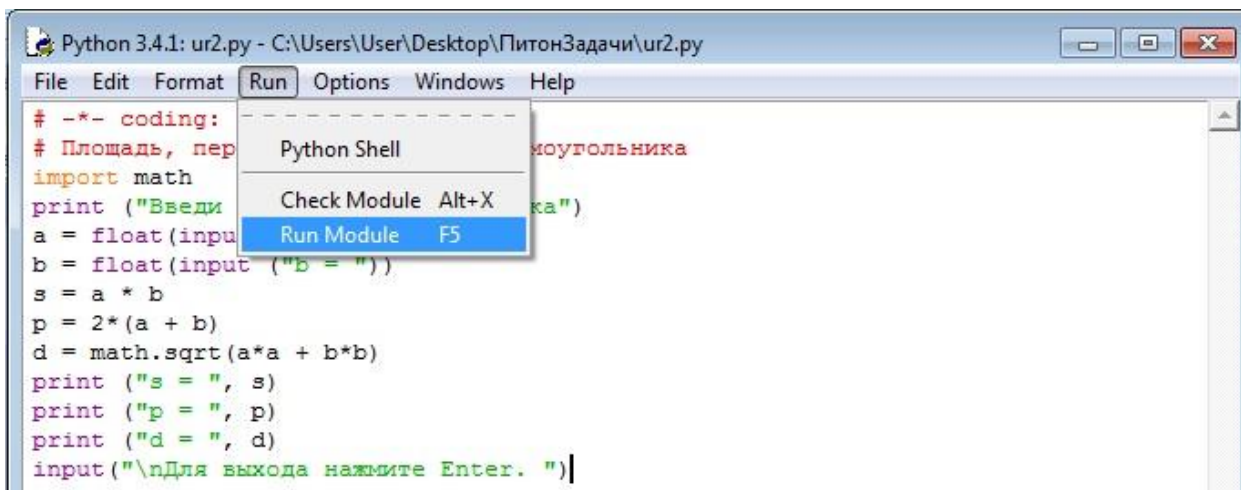


Рис 2.2. Запуск программы на выполнение из среды программирования IDLE Python 3

Результат выполнения программы из среды программирования IDLE Python 3 после ввода данных (рис 2.3).

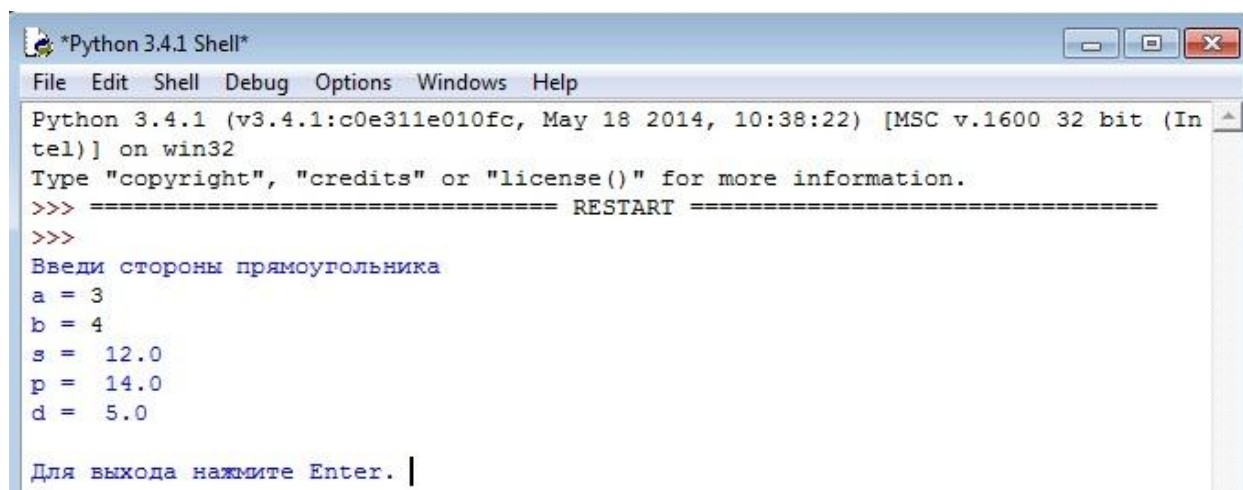


Рис 2.3. Результат выполнения программы в среде программирования

Открыть файл с кодом на Python из папки, где находятся файлы, можно с помощью контекстного меню значка файла, выбрав пункт меню Edit with IDLE (рис 2.4).

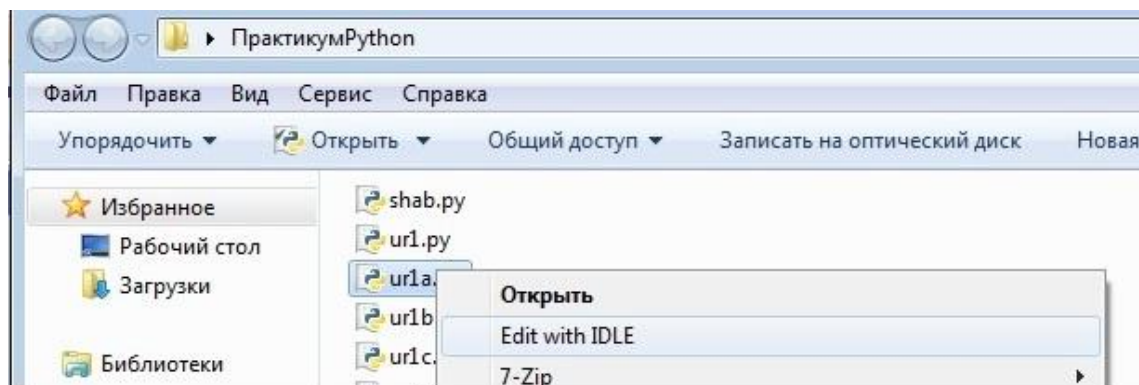


Рис 2.4. Открытие программы с помощью контекстного меню значка файла

Запустить программу на выполнение вне среды программирования можно двойным щелчком на значке программы (рис 2.5).

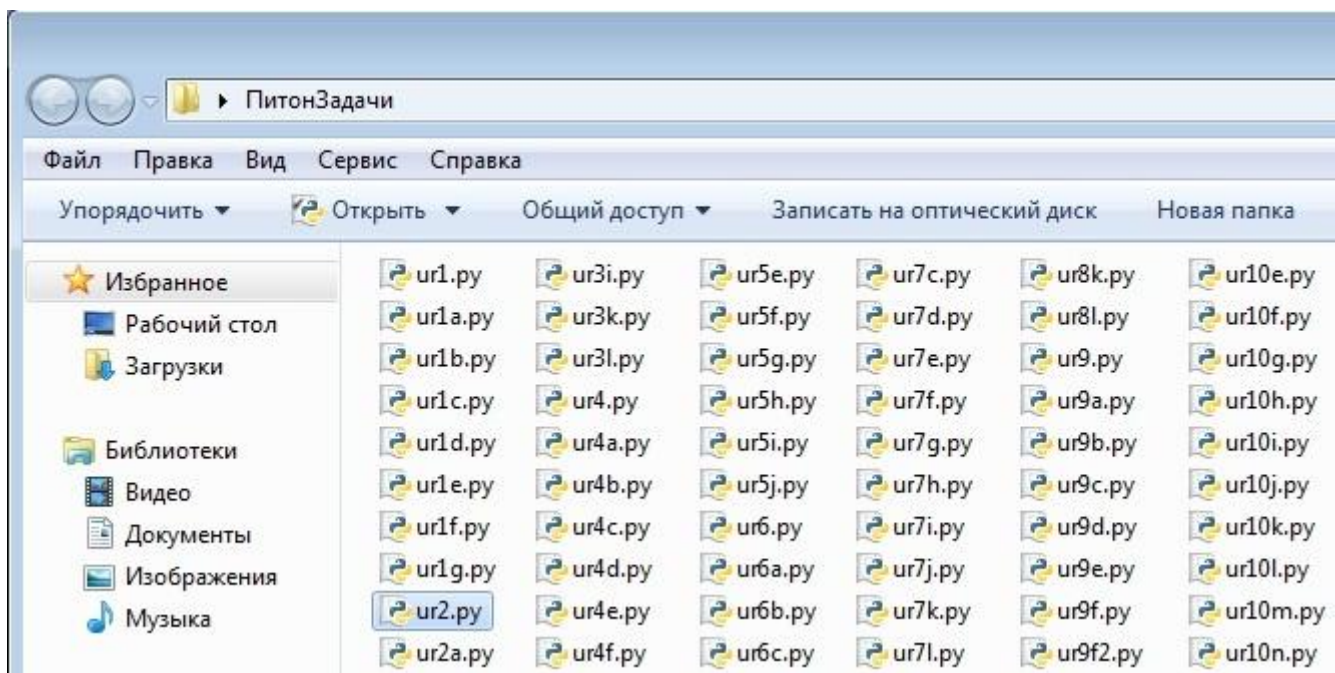


Рис 2.5. Запуск программы вне среды программирования

Программа будет запущена в текстовом окне консоли (рис 2.7). Консоль – это единое устройство, вводом для которого, является клавиатура, а выводом – монитор. Для графических операционных систем, вместо монитора, результат выводится в текстовое окно.

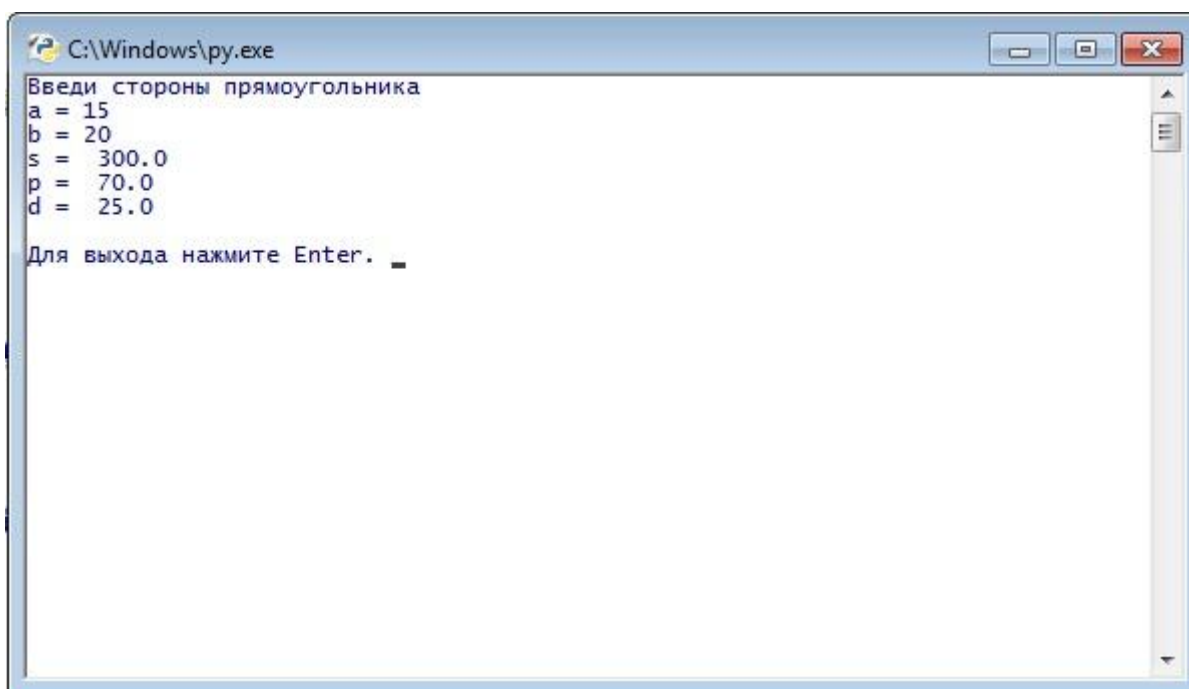


Рис 2.7. Результат выполнения программы в окне консоли

Распечатанный текст кода программы называется листингом. Большая часть времени программиста проходит над изучением листингов программ. По традиции, начинающий программист должен начинать с «первой программы программиста». Приводим код данной программы (листинг 2.1).

Листинг 2.1. Первая программа программиста

```
# -*- coding: utf-8 -*-
# Первая программа программиста
print ("Hello World!")
input("\nДля выхода нажмите Enter.")
```

Первая строка означает кодировку. Она должна быть, если в коде имеются буквы кириллицы – русский текст. Вторая строка является комментарием. Комментарий – пояснительный текст на английском или национальном языке. В третьей строке – функция вывода `print()`, аргументом которой, является строковая константа. Четвертая строка содержит функцию ввода `input()`. Она необходима, чтобы не закрылось окно консоли, после ввода данных, и мы смогли увидеть результат и проанализировать его. Окно консоли можно закрыть нажатием клавиши `Enter` или обычным способом. Аргументом функции является строка, содержащая после кавычек, знаки `"\n"`. Это символы экранированной последовательности. Эти знаки не выводятся на экран, а являются командами, выполняющими некоторые действия. В данном случае выполняется перевод курсора на новую строку. В табл. 2.1 приведены некоторые экранированные символы.

Таблица. 2.1 Экранированные символы

Название	Символы	Выполняет действие
Обратный слеш	<code>\\</code>	Выводит знак <code>\</code>
Апостроф	<code>\'</code>	Выводит знак <code>'</code>
Кавычка	<code>\"</code>	Выводит знак <code>"</code>
Звук системного динамика	<code>\a</code>	Издает звук системного динамика
Новая строка	<code>\n</code>	Переводит курсор на новую строку
Горизонтальная табуляция	<code>\t</code>	Перемещает курсор вправо на отступ

Аналогом «первой программы программиста» является программа, выводящая некоторое текстовое предложение.

Листинг 2.2. Аналог первой программы программиста

```
# -*- coding: utf-8 -*-
# Аналог первой программы программиста
print("Изучайте Python!")
input("\nДля выхода нажмите Enter!")
```

П2. Внимательно прочитайте текст заданий практикума и выполните их.

2.1. Создайте на Рабочем столе папку с именем `PrakPyt`. Выполните команду меню `File\ New File`. Сохраните файл под именем `shab.py`. В редакторе среды программирования наберите следующий текст кода и сохраните изменение в файле (листинг 2.1).

Листинг 2.3. Шаблон программы

```
# -*- coding: utf-8 -*-
#
input("\nДля выхода нажмите Enter!")
```

- 2.2. Откройте в редакторе кода IDLE Python 3 файл `shab.py`. Сохраните под новым именем `url.py`. Дополните код в файле, чтобы получился код «первой программы программиста» (листинг 2.2). Сохраните изменение в файле. Запустите программу из среды и вне среды программирования.

Листинг 2.4. Первая программа программиста

```
# -*- coding: utf-8 -*-
# Первая программа программиста
print("Hello world!")
input("\nДля выхода нажмите Enter!")
```

- 2.3. Откройте в редакторе кода IDLE Python3 файл `url.py`. Сохраните его под новым именем `urla.py`. Вторую и третью строки замените на:

```
# Первая программа программиста РФ
print("Привет, мир!")
```

Сохраните изменение в файле (листинг 2.3). Запустите программу из среды и вне среды программирования.

Листинг 2.5. Первая программа программиста РФ

```
# -*- coding: utf-8 -*-
# Первая программа программиста РФ
print("Привет, мир!")
input("\nДля выхода нажмите Enter!")
```

- 2.4. Создайте программу, выводящую фразу "Изучайте Python.". Сохраните под именем `urlb.py`. Запустите программу из среды и вне среды программирования.

Листинг 2.6. Аналог первой программы программиста

```
# -*- coding: utf-8 -*-
# Аналог первой программы программиста
print("Удивительный Python!")
input("\nДля выхода нажмите Enter!")
```

С 2. Создайте на Рабочем столе папку с именем `SamPyt`. Все файлы самостоятельных работ сохраняйте в этой папке. Откройте файл под именем `shab.py` и сохраните его в эту папку.

- 2.1. Откройте в редакторе кода IDLE Python 3 файл `shab.py`. Сохраните его под новым именем `sam1.py`. Дополните код в файле, чтобы получился код «первой программы программиста» (листинг 2.2). Сохраните изменение в файле. Запустите программу из среды и вне среды программирования.

- 2.2. Откройте в редакторе кода IDLE Python3 файл `sam1.py`. Сохраните его под новым именем `sam1a.py`. Вторую и третью строки замените на:

```
# Первая программа программиста РФ
print("Привет, всем программистам мира!")
```

Сохраните изменение в файле (листинг 2.3). Запустите программу из среды и вне среды программирования.

2.3. Создайте программу, выводящую фразу "Удивительный Python.". Сохраните под именем `sam1b.py`. Запустите программу из среды и вне среды программирования.

Р 2. Вы умеете работать в сценарном режиме среды программирования IDLE Python3. Научились создавать файл программы, редактировать его, запускать на выполнение. Создав свою «первую программу программиста», вы, приняты в сообщество программистов.

Л 2. Команда Run ▪ Команда Run\Module ▪ Сценарный режим ▪ Файл программы ▪ Код ▪ Листинг ▪ Кодировка ▪ Комментарий ▪ Первая программа программиста ▪ Экранированные символы ▪

Урок 3. Строковый тип данных

Ц 3. Дать первоначальные сведения о строковом типе данных, операторах конкатенации и повторения строк, функциями `input()` – ввода, `print()` – вывода данных.

К 3. Информация, обрабатываемая компьютером, должна быть представлена в виде данных. Данные в языке Python являются объектами. Это – числа, строки, логические значения, списки и другие объекты. Данные хранятся в памяти компьютера. Для доступа к объекту используются переменные. Переменные с помощью оператора присваивания получают значение. Начальное присваивание значения переменной называется инициализацией. Оператор присваивания имеет следующий синтаксис:

переменная = выражение

В данном синтаксисе *переменная* – это имя (идентификатор) переменной, знак "=" – оператор присваивания, *выражение* – константа, переменная или выражение, составленное из них, с использованием операторов и скобок.

Переменные в языке Python отличаются от переменных в других языках. В языках Pascal, C++ и других, переменным в начале программы указывается их тип, который, не может быть изменен дальше в программе. Такой способ переменных называется статическим. В Python заранее не нужно объявлять их тип. Тип переменной может меняться при работе программы. Переменная в Python не хранит данные, а ссылается на них. Тип переменной в Python является динамическим.

Строковая константа – последовательность символов, заключенные в кавычки или апострофы.

Примеры: "robot", "125", "информатика", 'Python', '2+3='.

Если в правой части оператора присваивания, находится строковое выражение, то переменная становится строковой. Над строковыми величинами определены операции конкатенации и повторения строк (табл. 3.1). Строки в Python являются сложным типом данных. Изучению их будет отведена отдельная глава. Здесь приводятся самые первоначальные сведения о строках.

Таблица. 3.1 Строковые операторы

Название	Символ	Пример	Результат
Конкатенация	+	<code>print("тепло" + "ход")</code>	теплоход
Повторение	*	<code>print("Байт"*3)</code>	БайтБайтБайт

Для ввода данных в Python используется функция `input()`, которая имеет следующий синтаксис:

переменная = input("текст")

В данном синтаксисе *переменная* – это имя переменной, знак "=" – оператор присваивания, `input()` – встроенная функция ввода данных, "текст" – подсказка пользователю, что он должен вводить.

П3. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

Листинг 3.1. Оператор конкатенации строк

```
# -*- coding: utf-8 -*-
# Оператор конкатенации строк
s = "Python"
print ("Я изучаю язык " + s + ".")
print (s + " – современный язык. ")
print ("Мне нравится язык " + s + ".")
input("\nДля выхода нажмите Enter.")
```

Листинг 3.2. Оператор повторения строк

```
# -*- coding: utf-8 -*-
# Оператор повторения строк
print("Изучаем Python!"*5)
input("\nДля выхода нажмите Enter!")
```

Листинг 3.3. Ввод/вывод строковых данных

```
# -*- coding: utf-8 -*-
# Ввод/вывод строковых данных
# Диалог работа о Python
s1 = "Привет"
s2 = "Python"
name = ""
print("Я – робот!")
name = input("Введи свое имя: ")
print(s1 + ", " + name + "!")
print (name + ", " + "изучай язык " + s2 + ".")
print (s2 + " – современный язык.")
```

Листинг 3.4. Диалог работа об одноклассниках

```
# -*- coding: utf-8 -*-
# Ввод/вывод строковых данных
# Диалог работа об одноклассниках
s1 = "Привет"
print("Я – робот!")
name1 = input("Введи свое имя: ")
print(s1 + ", " + name1 + " !")
name2 = input("Введите имя своего одноклассника!")
print (name1 + " и " + name2 + " – одноклассники!")
input("\nДля выхода нажмите Enter.")
```

Листинг 2.4. Диалог работа о хобби

```

# -*- coding: utf-8 -*-
# Ввод/Вывод строковых данных
# Диалог робота о хобби
s1 = "Привет"
print("Я - робот!")
name = input("Введи свое имя: ")
print(s1 + ", " + name + " !")
hob = input("Назови свое хобби!")
print (hob + " - интересное занятие.")
print (name + ", не бросай " + hob + "!" )
input("\nДля выхода нажмите Enter.")

```

С 3. Внимательно прочитайте тексты заданий. Откройте в редактор кода IDLE Python 3. Составьте программы для них файлы программ и выполните их для различных входных данных.

- 3.1. Составьте программу, запрашивающую у пользователя две строки и выводящую конкатенацию этих строк.
- 3.2. Составьте программу, запрашивающую у пользователя строку и выводящую 5 раз эту строку с пробелом между ними.
- 3.3. Составьте программу, запрашивающую у пользователя строку и выводящую 4 раза эту строку в столбик.
- 3.4. Составьте программу диалога робота с пользователем о поэтах России.
- 3.5. Составьте программу диалога робота с пользователем о языках программирования.
- 3.6. Составьте программу диалога робота с пользователем о спорте.

Р 3. Вы знаете, что такое строковый тип данных, и какие операции можно выполнять над строками. Умеете применять функции ввода и вывода данных. Составляете разнообразные диалоговые программы.

Л 3. Функция input() ▪ Функция print() ▪ Строковый тип ▪ Конкатенация ▪ Повторение строк ▪ Оператор присваивания ▪

Урок 4. Целочисленный тип данных

Ц 4. Дать первоначальные сведения о целочисленном типе данных, функции int() – перевода строки в целое число, математических операторах целых чисел.

К 4. Математические операторы целых чисел в Python соответствуют математическим операциям с целыми числами в математике. В табл. 4.1 приведены математические операторы целых чисел.

Таблица 4.1. Математические операторы целых чисел

Название	Символ	Пример	Результат
Сложение	+	5 + 8	13
Вычитание	-	16 - 9	7
Умножение	*	15 * 4	60
Деление	/	20 / 4	5.0
Степень	**	2 ** 5	32
Целочисленное деление	//	17 // 3	5
Остаток от деления	%	17 % 3	2

Результат при делении целых чисел является вещественным числом, для остальных операторов результат – целое число.

П 4. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

Листинг 4.1. Сумма двух целых чисел

```
# -*- coding: utf-8 -*-
# Сумма двух целых чисел
a = 5
b = 7
s = a + b
print ("a = ", a)
print ("b = ", b)
print ("a + b =", s)
input("\nДля выхода нажмите Enter. ")
```

Листинг 4.2. Неправильное вычисление суммы двух целых чисел

```
# -*- coding: utf-8 -*-
# Неправильное вычисление суммы двух целых чисел
# По умолчанию аргумент функции
# Является строковой величиной
print ("Введи целые числа!")
a = input("a = ")
b = input("b = ")
s = a + b
print ("a + b = ", s)
print ("Результат неверный!")
input("\nДля выхода нажмите Enter. ")
```

Листинг 4.3. Функция int() – перевод строки в целое число

```
# -*- coding: utf8 -*-
# Правильное вычисление суммы двух целых чисел
# Функция int()- перевод строки в целое число
print ("Введи целые числа!")
a = int(input("a = "))
b = int(input("b = "))
s = a + b
print ("a + b = ", s)
print ("Результат верный!")
input("\nДля выхода нажмите Enter. ")
```

Листинг 4.4. Целочисленное деление

```
# -*- coding: utf-8 -*-
# Целочисленное деление
print ("Введи целые числа!")
a = int(input("a = "))
```

```

b = int(input("b = "))
d = a // b
print ("a // b = ", d)
input("\nДля выхода нажмите Enter. ")

```

Листинг 4.5. Остаток от деления

```

# -*- coding: utf-8 -*-
# Остаток от деления
print ("Введи целые числа!")
a = int(input("a = "))
b = int(input("b = "))
d = a % b
print ("a % b = ", d)
input("\nДля выхода нажмите Enter. ")

```

Листинг 4.6. Степень числа

```

# -*- coding: utf-8 -*-
# Степень числа
print ("Введи целые числа!")
a = int(input("Основание a = "))
b = int(input("Показатель b = "))
d = a ** b
print ("a ** b = ", d)
input("\nДля выхода нажмите Enter.")

```

Листинг 4.7. Сумма, произведение цифр двузначного числа

```

# -*- coding: utf-8 -*-
# Сумма, произведение цифр двузначного числа
a = int(input("Введи двузначное натуральное число!"))
e = a%10      # цифра единиц
d = a //10    # цифра десятков
s = d + e     # сумма цифр
p = d*e       # произведение цифр
print ("Сумма цифрр числа = ", s)
print ("Произведение цифрр числа = ", p)
input("\nДля выхода нажмите Enter. ")

```

Листинг 4.8. Реверс трехзначного числа

```

# -*- coding: utf-8 -*-
# Реверс трехзначного числа
n = int(input("Введи трехзначное натуральное число!"))
e = n%10      # цифра единиц
d = (n //10)%10 # цифра десятков
c = n//100    # цифра сотен
n = 100*e + 10*d + c # реверсное число
print ("Реверсное число = ", n)
input("\nДля выхода нажмите Enter. ")

```


С 4. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

4.1. Откройте в редакторе кода IDLE Python 3 файл программы практикума, в котором вычисляется сумма двух чисел. Числа вводятся с клавиатуры. Сохраните его под именем sam2.py. Измените программу так, чтобы она вычисляла сумму квадратов двух чисел.

4.2. Откройте в редакторе кода IDLE Python 3 файл программы sam2.py. Сохраните его под именем sam2a.py. Измените программу так, чтобы она вычисляла квадрат суммы двух чисел.

4.3. Составьте программу, вычисляющую степень числа 2. Программа должна запрашивать ввод показателя.

4.4. Составьте программу, вычисляющую сумму квадратов крайних чисел трехзначного числа.

4.5. Составьте программу, вычисляющую объем, сумму ребер, и площадь полной поверхности куба.

4.6. Составьте программу, вычисляющую объем, сумму ребер, и площадь полной поверхности прямоугольного параллелепипеда.

Р 4. Вы умеете составлять программы с целыми числами в Python. Применяете функцию int() для преобразования строкового типа в целочисленный тип данных. Уверенно применяете математические операторы целых чисел.

Л 4. Функция int() ▪ Целочисленный тип ▪ Математические операторы целых чисел ▪ Целочисленное деление ▪ Остаток от деления ▪

Урок 5. Тип данных вещественных чисел

Ц 5. Дать первоначальные сведения о вещественных числах в Python, функции float() – перевода строки в вещественное число, математических операторов вещественных чисел.

К 5. Математические операторы вещественных чисел в Python соответствуют математическим операциям с десятичными дробями в математике. В табл. 4.1 приведены математические операторы вещественных чисел.

Таблица 5.1 Математические операторы вещественных чисел

Название	Символ	Пример	Результат
Сложение	+	16.23+34.58	50.81
Вычитание	-	56.8-23.15	33.65
Умножение	*	2.5*2.8	7.0
Деление	/	10.24/1.28	8.0
Степень	**	2.5**4	39.0625

П 5. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

Листинг 5.1. Сумма двух вещественных чисел

```
# -*- coding: utf-8 -*-
# Сумма двух вещественных чисел
a = 5.25
b = 17.452
```

```

s = a + b
print ("a = ", a)
print ("b = ", b)
print ("a + b = ", s)
input("\nДля выхода нажмите Enter. ")

```

Листинг 5.2. Неправильное вычисление суммы двух вещественных чисел

```

# -*- coding: utf-8 -*-
# Неправильное вычисление
# суммы двух вещественных чисел
# По умолчанию аргументом функции input() является строка
print ("Введи вещественные числа!")
a = input("a = ")
b = input("b = ")
s = a + b
print ("a + b = ", s)
print ("Результат неверный!")
input("\nДля выхода нажмите Enter. ")

```

Листинг 5.3. Функция float() – перевод строки в вещественное число

```

# -*- coding: utf-8 -*-
# Сумма двух вещественных чисел
# По умолчанию аргумент функции input()
# является строковой величиной
# Функция float() преобразования
# строки в вещественное число
print ("Введи вещественные числа!")
a = float(input("a = "))
b = float(input("b = "))
s = a + b
print ("a + b = ", s)
print ("Результат верный!")
input("\nДля выхода нажмите Enter. ")

```

Листинг 5.4. Обмен значениями переменными через буфер

```

# -*- coding: utf-8 -*-
# Обмен значениями переменными через буфер
print ("Введи числа!")
a = float(input("a = "))
b = float(input("b = "))
buf = b
b = a
a = buf
print ("Значения переменных a и b после замены")
print ("a = ", a)
print ("b = ", b)
input("\nДля выхода нажмите Enter. ")

```

Листинг 5.5. Обмен значениями переменными через присваивание

```
# -*- coding: utf-8 -*-
# Обмен значениями переменными через присваивание
print ("Введи числа!")
a = float(input("a = "))
b = float(input("b = "))
a,b = b,a
print ("Значения переменных a и b после замены")
print ("a = ", a)
print ("b = ", b)
input("\nДля выхода нажмите Enter. ")
```

Листинг 5.6. Округление вещественных чисел

```
# -*- coding: utf-8 -*-
# Округление вещественных чисел
a = 3425.4567202
b = 0.0075678
c = -2345.450986
d = -2.0004561
print ("Данные числа")
print ("a = ", a)
print ("b = ", b)
print ("c = ", c)
print ("d = ", d)
a = round(a,3)
b = round(b,4)
c = round(c,2)
d = round(d,5)
print ("Округленные числа ")
print ("a = ", a)
print ("b = ", b)
print ("c = ", c)
print ("d = ", d)
input("\nДля выхода нажмите Enter. ")
```

Листинг 5.7. Форматированный вывод вещественных чисел

```
# -*- coding: utf-8 -*-
# Форматированный вывод вещественных чисел
a = 3425.4567202
b = 0.00045678
print ("Данные числа")
print ("a = ", a)
print ("b = ", b)
print ("Отформатированные числа ")
print ("a = ", "%f" %a)
print ("a = ", "%g" %a)
print ("a = ", "%e" %a)
print ("b = ", "%f" %b)
print ("b = ", "%g" %b)
```

```
print ("b = ", "%e" %b)
input("\nДля выхода нажмите Enter. ")
```

С 5. Откройте редактор кода IDLE Python 3. Составьте программы вычисления величин, данных в условии задания. Протестируйте программы для разных входных данных.

5.1. Составьте программу, вычисляющую сумму кубов двух чисел. Результат округлите до двух знаков в дробной части числа.

5.2. Составьте программу, вычисляющую куб разности двух чисел. Результат округлите до трех знаков в дробной части числа.

5.3. Составьте программу, вычисляющую степень числа 2.5. Программа должна запрашивать ввод показателя.

5.4. Составьте программу, вычисляющую среднее арифметическое трех вещественных чисел. Числа вводятся с клавиатуры. Результат округлите до двух знаков в дробной части числа.

5.5. Составьте программу, вычисляющую периметр равнобедренной трапеции. Снования и боковая сторона вводятся с клавиатуры. Результат округлите до одного знака в дробной части числа.

5.6. Составьте программу, вычисляющую объем, сумму ребер, и площадь полной поверхности прямоугольного параллелепипеда. Результат округлите до одного знака в дробной части числа.

Р 5. Вы умеете составлять программы с вещественными числами в Python. Применяете функцию float() для преобразования строкового типа в вещественный тип данных. Уверенно применяете математические операторы вещественных чисел.

Л 5. ▪ Функция float() ▪ Вещественный тип ▪ Округление чисел ▪ Форматированный вывод ▪

Урок 6. Способы подключения модуля

Ц 6. Познакомить со способами подключения модуля в Python.

К 6. Язык Python поддерживает модульную парадигму программирования. Парадигма языка – это принцип, положенный в его конструкцию. Модульный принцип подразумевает разбивку сложной программы на отдельные части – модули. Среда программирования IDLE Python3 – это сложная программа, а язык Python – язык современный, поэтому в его основу положен модульный принцип. В модуль, обычно, помещаются подпрограммы для работы с определенными данными. Например, в модуле math, содержатся алгебраические и тригонометрические функции и константы для работы с вещественными числами. Прежде, чем применить функции модуля, его нужно подключить. В табл. 6.1 приведены четыре способа подключения математического модуля math.

Таблица. 6.1 Способы подключения математического модуля math

Способ подключения	Синтаксис	Пример	Результат
Подключение модуля math	import math	math.sqrt(4)	2.0
Подключение модуля math через псевдоним	import math as m	m.sqrt(16)	4.0
Подключение только функции sqrt модуля math	from math import sqrt	sqrt(6.25)	2.5
Подключение модуля math без использования его имени в коде	from math import *	sqrt(144)	12.0

П 6. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

Листинг 6.1. Подключение модуля math

```
# -*- coding: utf-8 -*-
# Подключение модуля math
# Корень квадратный из числа
import math      # подключаем модуль
a = float(input("Введи число a = "))
n = math.sqrt(a) # вычисляем модуль
n = round(n,2)   # округляем до двух знаков дробной части
print ("Квадратный корень из числа ", a, " равен ",n)
input("\nДля выхода нажмите Enter. ")
```

Листинг 6.2. Подключение модуля через псевдоним

```
# -*- coding: utf-8 -*-
# Подключение модуля через псевдоним
# Корень квадратный из числа
import math as m # подключаем модуль через псевдоним
a = float(input("Введи число a = "))
n = m.sqrt(a)
n = round(n,1)   # округляем до одного знака дробной части
print ("Квадратный корень из числа ", a, " равен ",n)
input("\nДля выхода нажмите Enter. ")
```

Листинг 6.3. Подключение функции sqrt модуля math

```
# -*- coding: utf-8 -*-
# Подключение определенной функции модуля
# Корень квадратный из числа
from math import sqrt
a = float(input("Введи число a = "))
n = sqrt(a)
n = round(n,1)
print ("Квадратный корень из числа ", a, " равен ",n)
input("\nДля выхода нажмите Enter. ")
```

Листинг 6.4. Подключение модуля math без его имени в коде

```
# -*- coding: utf-8 -*-
# Подключение модуля без его имени в коде
# Корень квадратный из числа
from math import *
a = float(input("Введи число a = "))
n = sqrt(a)
n = round(n,2)
print ("Квадратный корень из числа ", a, " равен ",n)
```

```
input("\nДля выхода нажмите Enter. ")
```

С 6. Используя различные способы подключения модуля `math`, составьте следующие программы.

6.1. Используя подключение модуля `math`, составьте программу, вычисляющую длину окружности и площадь круга по его радиусу.

6.2. Используя подключение модуля `math` модуля через псевдоним, составьте программу, вычисляющую объем и площадь поверхности шара по его радиусу.

6.3. Используя подключение функции `sqrt` модуля `math`, составьте программу, вычисляющую среднее геометрическое двух чисел.

6.4. Используя подключение функции `sqrt` модуля `math`, составьте программу, вычисляющую площадь треугольника по трем его сторонам.

6.5. Используя подключение модуля `math` без использования его имени в коде, составьте программу, вычисляющую гипотенузу прямоугольного треугольника по двум его катетам.

6.6. Используя подключение модуля `math` без использования его имени в коде, составьте программу, вычисляющую катет прямоугольного треугольника по проекции этого катета и гипотенузе.

Р 6. Вы умеете четырьмя способами подключать математический модуль `math`, и использовать его функции для решения разнообразных вычислительных задач.

Л 6. ▪ Модуль `math` ▪ `import math` ▪ `import math as m` ▪ `from math import sqrt` ▪ `from math import *` ▪ Псевдоним ▪
▪ Функции модуля ▪ Псевдоним модуля ▪

Урок 7. Математические функции модуля `math`

Ц 7. Познакомить с математическими функциями модуля `math`

К 7. В табл. 7.1 приведены тригонометрические функции, а в табл. 7.2 алгебраические функции математического модуля `math`. Прежде чем использовать функции модуля, его необходимо подключить одним из четырех способов, рассмотренных в предыдущем уроке.

Таблица 7.1 Тригонометрические функции модуля `math`

Название	Имя	Пример	Результат
Константа пи	<code>pi</code>	<code>math.pi</code>	3.141592653589793
Синус	<code>sin()</code>	<code>math.sin(pi/6)</code>	0.5
Косинус	<code>cos()</code>	<code>math.cos(pi)</code>	-1.0
Тангенс	<code>tan()</code>	<code>math.tan(pi/4)</code>	1.0
Арксинус	<code>asin()</code>	<code>math.asin(30)</code>	0.55
Аркосинус	<code>acos()</code>	<code>math.acos(30)</code>	1.02
Арктангенс	<code>atan()</code>	<code>math.atan(30)</code>	0.48
Перевод градусов в радианы	<code>radians()</code>	<code>math.radians(57.3)</code>	1.0000736613927508

Таблица 7.2 Математические функции модуля `math`

Название	Имя	Пример	Результат
Константа <code>e</code>	<code>e</code>	<code>math.e</code>	2.71828182845904
Модуль числа	<code>fabs()</code>	<code>math.fabs(-20.6)</code>	20.6

Округление до большего целого числа	ceil()	math.ceil(12.3)	13
Округление до меньшего целого числа	floor()	math.floor(12.8)	12
Степень по основанию e	exp()	math.exp(e)	15.154262241479262
Логарифм по основанию e	log()	math.log(e)	1.0
Логарифм по основанию 2	log2()	math.log2(128)	7.0
Логарифм по основанию 10	log10()	math.log10(1000)	3.0

П 7. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

Листинг 7.1. Тригонометрические функции модуля math

```
# -*- coding: utf-8 -*-
# Тригонометрические функции
import math
print ("Данные числа")
pi = math.pi
print ("pi = ", pi)
a = round(math.sin(pi/6),1)
print ("sin(pi/6) = ", a)
b = round(math.cos(pi),1)
print ("cos(pi) = ", b)
c = round(math.tan(pi/4),1)
print ("tan(pi/4) = ", c)
input("\nДля выхода нажмите Enter. ")
```

Листинг 7.2. Функция radians() преобразования градусов в радианы

```
# -*- coding: utf-8 -*-
# Преобразование градусов в радианы
import math
print ("Введи угол в градусах!")
a = int(input("a = "))
a = math.radians(a)
a1 = math.sin(a)
a1 = round(a1,2)
b = math.cos(a)
b1 = round(b,2)
c = math.tan(a)
c = round(c,2)
print ("sin(a) = ", a1)
print ("cos(a) = ", b1)
print ("tan(a) = ", c)
input("\nДля выхода нажмите Enter. ")
```

Листинг 7.3. Математические функции модуля math

```
# -*- coding: utf-8 -*-
# Математические функции
```

```

import math
print ("Данные числа")
e = math.e
print ("e = ", e)
a = math.exp(e)
print ("exp(e) = ", a)
b = math.log(e)
print ("log(e) = ", b)
c = math.log2(128)
print ("log2(128) = ", c)
d = math.log10(1000)
print ("log10(1000) = ", d)
f = math.fabs(-20.6)
print ("fabs(-20.6) = ", f)
g = math.ceil(12.3) # округление до большего целого
print ("ceil(12.3) = ", g)
h = math.floor(12.8) # округление до меньшего целого
print ("floor(12.8) = ", h)
input("\nДля выхода нажмите Enter. ")

```

С 7. Составьте программы вычисления величин, данных в условии задания. Протестируйте программы для разных входных данных.

7.1. Составьте программу вычисления абсолютной величины числа. Число вводится с клавиатуры.

7.2. Составьте программу округления до большего целого, до меньшего целого десятичной дроби. Число вводится с клавиатуры.

7.3. Составьте программу вычисления функции $\exp(x)$. Число x вводится с клавиатуры.

7.4. Составьте программу вычисления функции $\log(x)$. Число x вводится с клавиатуры.

7.5. Составьте программу вычисления функции $\log_2(x)$. Число x вводится с клавиатуры.

7.6. Составьте программу вычисления функции $\log_{10}(x)$. Число x вводится с клавиатуры.

Р 7. Вы умеете составлять программы с вещественными числами в Python, использующие тригонометрические и алгебраические функции математического модуля `math`.

Л 7. ▪ Функция `float()` ▪ Вещественный тип ▪ Округление чисел ▪ Форматированный вывод ▪

Урок 8. Программы линейной структуры

Ц 8. Познакомить с алгоритмом составления линейных программ и применения его для решения вычислительных задач.

К 8. Программа имеет линейную структуру, если все ее инструкции выполняются последовательно друг за другом. Линейная программа не содержит ни ветвлений, ни циклов, ни подпрограмм. Для языка Python все инструкции линейной программы должны начинаться сначала. В начале строки не должно быть пробела. Все предыдущие программы были линейными. Одной из разновидностей линейных программ являются программы вычисления некоторых величин. Прежде чем составлять код таких программы, необходимо внимательно прочитать текст задания и выделить входные величины, которые даны в условии. Дать имена таким переменным и организовать их ввод с

помощью функции `input()`. Записать формулы, по которым вычисляются величины, которые нужно вычислить. Вывести требуемые величины, используя функцию вывода `print()`. Запустив программу на выполнение, нужно проверить ее работу на данных, результат для которых вам известен. Программа может работать, но выдавать неверный результат. Возможно, использована не та формула для вычисления величин.

П 8. Откройте в редакторе кода IDLE Python 3 файлы программ практикума и выполните их для различных входных данных. Внимательно читайте комментарии и анализируйте текст кода.

Листинг 8.1. Периметр, площадь, диагональ прямоугольника

```
# -*- coding: utf-8 -*-
# Периметр, площадь, диагональ прямоугольника
import math
print ("Введи стороны прямоугольника!")
a = float(input("a = "))
b = float(input("b = "))
p = 2*(a + b)
s = a*b
d = math.sqrt(a*a + b*b)
print ("p = ", p)
print ("s = ", s)
print ("d = ", d)
input("\nДля выхода нажмите Enter. ")
```

Листинг 8.2. Сумма цифр трехзначного числа

```
# -*- coding: utf-8 -*-
# Сумма цифр трехзначного числа
n = int(input("Введи трехзначное натуральное число: "))
e = n % 10      # цифра единиц
d = (n//10)%10 # цифра десятков
c = n//100     # цифра сотен
s = c + d + e  # сумма цифр
print ("Сумма цифр s = ", s)
input("\nДля выхода нажмите Enter. ")
```

Листинг 8.3. Площадь треугольника по сторонам и синусу между ними

```
# -*- coding: utf-8 -*-
# Площадь треугольника по сторонам и синусу между ними
# Мера угла в градусах
import math
a = float(input("Введи сторону a = "))
b = float(input("Введи сторону b = "))
c = int(input("Введи угол в градусах c = "))
c = math.radians(c)
s = a*b*math.sin(c)/2
s = round(s,2) # два знака в дробной части
print ("Площадь треугольника s = ", s)
input("\nДля выхода нажмите Enter. ")
```

Листинг 8.4. Среднее арифметическое и геометрическое

```
# -*- coding: utf-8 -*-
# Среднее арифметическое
# Среднее геометрическое
import math
a = float(input("Введи число a = "))
b = float(input("Введи число b = "))
sar = (a + b)/2
sgeom = math.sqrt(a * b)
sar = round(sar,1) # один знак после запятой
sgeom = round(sgeom,1)
print ("Среднее арифметическое = ", sar)
print ("Среднее геометрическое = ", sgeom)
input("\nДля выхода нажмите Enter. ")
```

Листинг 8.5. Потенциальная и кинетическая энергия

```
# -*- coding: utf-8 -*-
# Потенциальная и кинетическая энергия
g = 9.8
m = float(input("Введи m = "))
h = float(input("Введи h = "))
v = float(input("Введи v = "))
Ep = m*g*h
Ek = m*v*v/2
Ep = round(Ep,2) # два знака после запятой
Ek = round(Ek,2)
print ("Потенциальная энергия = ",Ep )
print ("Кинетическая энергия = ",Ek )
input("\nДля выхода нажмите Enter. ")
```

Листинг 8.6. Успеваемость в классе по предмету

```
# -*- coding: utf-8 -*-
# Успеваемость в классе по предмету
print ("Вводим данные об успеваемости")
n5 = int(input("На '5' = "))
n4 = int(input("На '4' = "))
n3 = int(input("На '3' = "))
n2 = int(input("На '2' = "))
n = n5 + n4 + n3 + n2 # вычисляем всего учащихся
# Вычисляем проценты по оценкам
z5 = (n5/n)*100
z4 = (n4/n)*100
z3 = (n3/n)*100
z2 = (n2/n)*100
usp = (n - n2)/n*100 # процент успеваемости
neusp = 100 - usp # процент неуспеваемости
kach = z5 + z4 # вычисляем качество знаний
srbal = (5*n5+4*n4+3*n3+2*n2)/n # вычисляем средний балл
# Округляем - один знак в дробной части
z5 = round(z5,1)
```

```

z4 = round(z4,1)
z3 = round(z3,1)
z2 = round(z2,1)
usp = round(usp,1)
neusp= round(neusp,1)
kach = round(kach,1)
srbal = round(srbal,1)
print ("На '5' = ", z5, "%")
print ("На '4' = ", z4 , "%")
print ("На '3' = ", z3 , "%")
print ("На '2' = ", z2 , "%")
print ("Успеваемость = ", usp, "%")
print ("Не успевают: = ", neusp , "%")
print ("Качество знаний: = ", kach , "%")
print ("Средний бал: ",srbal)
input("\nДля выхода нажмите Enter. ")

```

С 8. Составьте программы вычисления величин, данных в условии задания. Протестируйте программы для разных входных данных.

8.1. Составьте программу вычисления площади трапеции по основаниям a и b высоте h .

8.2. Составьте программу вычисления радиуса вписанной и описанной окружности равностороннего треугольника по его стороне a .

8.3. Составьте программу вычисления кольца, где r_1 и r_2 – радиусы внешней и внутренней окружностей.

8.4. Составьте программу вычисления объема и площади полной поверхности цилиндра по его радиусу r и высоте h .

8.5. Составьте программу вычисления объема и площади полной поверхности правильной пирамиды по ее стороне основания a и высоте h .

8.6. Составьте программу вычисления процентного вхождения значения каждой из величин для выражения $a + b - c$.

Р 8. Вы умеете применять алгоритм составления линейных программ для решения вычислительных задач.

Л 8. ▪ Линейная программа ▪ Тестирование программы ▪ Алгоритм решения вычислительной задачи ▪

