

ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ



Python — это высокоуровневый язык программирования, который используется в различных сферах IT, таких как машинное обучение, разработка приложений, web, парсинг и другие.

В 2019 году Python стал самым популярным языком программирования, обогнав Java на 10%. Это обусловлено многими причинами, одна из которых — высокая оплата труда квалифицированных специалистов (около 100 тысяч долларов в год).

Язык программирования Python

Различные языки программирования обычно доминируют в какой-то отрасли (или нескольких), для работы в которой они хорошо подходят. Но это не значит, что программист ограничен использовать строго определённый инструмент, поэтому любой язык общего назначения, такой как Python, может применять для создания чего-угодно.

Python смог захватить малую часть рынка веб-разработки, иногда используется для написания десктопных приложений и, конечно, тотально доминирует в сфере машинного обучения. Кроме того, на нём создаётся много прототипов, которые позволяют быстро набросать функционал и внешний вид будущего проекта.

Происхождение названия

Автор языка **Python** назвал его в честь британского комедийного шоу “**Monty Python**”, которое было популярно в начале 1970-х годов.

Это телешоу позволяло автору расслабиться и отвлечься от разработки языка. Однако, несмотря на настоящее происхождение названия, для людей более очевидно связывать Python со словом “змея”. Этому также способствует логотип, на котором изображена рептилия.

И хотя создатель языка не раз говорил, что название никак не связано со змеями, повлиять на мнение общества так и не удалось.

Питон или Пайтон?

Будь то название британского телешоу или английское звучание слова “змея”, **Python** **правильно произносить, как Пайтон**. Однако, около 80% Российского сообщества привыкли использовать слово “Питон”.

Нельзя сказать, что однозначно правильно использовать один из вариантов, многие названия адаптируются под произношения конкретного языка, а изменить сложившиеся привычки общества очень сложно. Однако, вариант названия “Питон” уместно употреблять только в разговоре с русскоязычными собеседниками, потому что на любой международной конференции значение слова “Питон” просто не поймут, ведь в английском языке его нет, есть только “Python (Пайтон)”.

Логотип

На логотипе изображены две змеи, образующие квадрат с выпуклым центром, это часто вводит в заблуждение пользователей, вынуждая ассоциировать название языка с рептилией.



Логотип создал брат автора, Юст ван Россум — программист и шрифтовой дизайнер. Он разработал как дизайн логотипа (две змеи), так и шрифт текста Flux Regular.

История создания

Язык начал разрабатывать программист, Гвидо ван Россумом, в конце 1980-х. На тот момент он работал в центре математики и информатике в Нидерландах.

Гвидо ван Россум увлекался работой с «железками» ещё со школьных лет, и хотя он не находил поддержки и одобрения у своих сверстников, это не помешало ему самостоятельно разработать язык программирования.

Философия (Дзен) языка Python.

Разработчики языка Python придерживаются определённой философии программирования, называемой "The Zen of Python". Её текст выдаётся интерпретатором Python по команде `import this`

Текст философии:

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если они не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один и, желательно, только один очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец.
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем прямо сейчас.
- Если реализацию сложно объяснить, то это плохая идея.
- Если реализацию легко объяснить, то идея, возможно хороша.
- Пространства имён - отличная штука! Будем делать их больше!



000.py x

```
1 import this
2
```

Shell x

>>> %Run 000.py

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

Основные типы данных

Имя	Тип	Описание
Целые числа	int	1, 2, 3
Числа с плавающей точкой	float	1.1, 2.4, 3.0
Строки	str	последовательность символов: "Hello", 'hello', "42"
Списки	list	последовательность объектов: [1, 2.0, "hello"]
Множества	set	последовательность уникальных объектов: {"a", "b"}
Булевы значения	bool	логические значения: True или False



Математические операции

Математическая запись	На языке Python	Результат операции
$2,5 + 0,5$	<code>2.5 + 0.5</code>	3
$2 + 5$	<code>2 + 5</code>	7
$10 - 8$	<code>10 - 8</code>	2
$3 \cdot 4$	<code>3 * 4</code>	12
$15 : 4$	<code>15 / 4</code>	3.75
Целая часть при делении	<code>15 // 4</code>	3
Остаток от деления	<code>13 % 4</code>	1
2^4	<code>2 ** 4</code>	16
$ -5 $	<code>abs(-5)</code>	5

Часто используют сокращенную запись арифметических операций:

Сокращенная запись	Полная запись
$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a /= b$	$a = a / b$

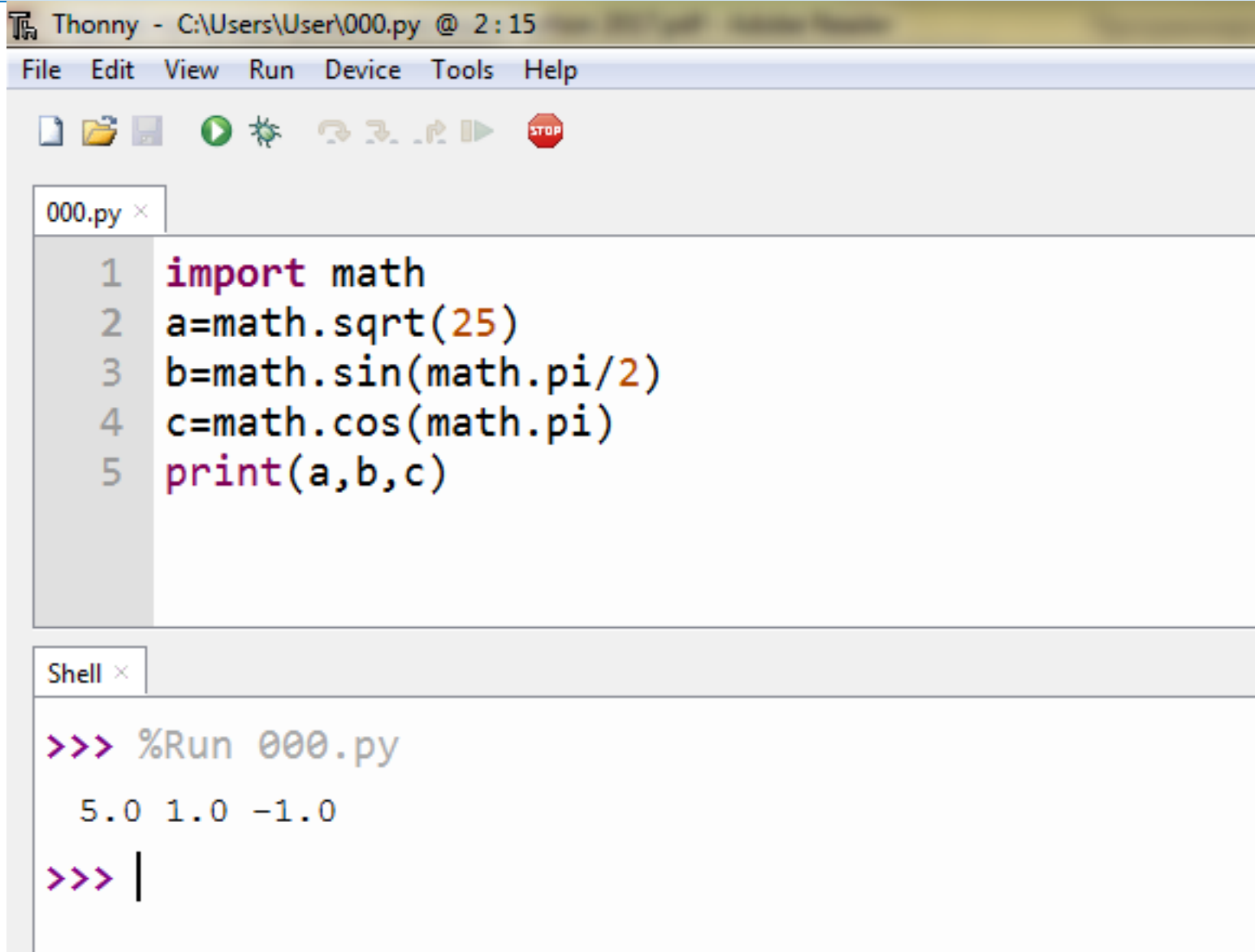
Вычисление квадратного корня можно заменить операцией возведения в степень $1/2$

$36^{}(1/2)$** или **$36^{**}0.5$**

Для вычисления дополнительных функций необходимо подключить модуль **math**.

Математическая запись	На языке Python
$\sqrt{25}$	<code>math.sqrt(25)</code>
$\sin(\pi)$	<code>math.sin(math.pi)</code>
$\cos(2\pi)$	<code>math.cos(2*math.pi)</code>
$\text{tg}(\pi/2)$	<code>math.tan(math.pi/2)</code>

Пример программы с подключением модуля `math`



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path is `C:\Users\User\000.py` and the time is 2:15. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for file operations and execution. The editor window, titled `000.py`, contains the following Python code:

```
1 import math
2 a=math.sqrt(25)
3 b=math.sin(math.pi/2)
4 c=math.cos(math.pi)
5 print(a,b,c)
```

The Shell window, titled `Shell`, shows the execution of the script using the `%Run` command:

```
>>> %Run 000.py
5.0 1.0 -1.0
>>> |
```

Команда «ВВОД ДАННЫХ»

```
a=input()
```

Ввод значения строковой
переменной

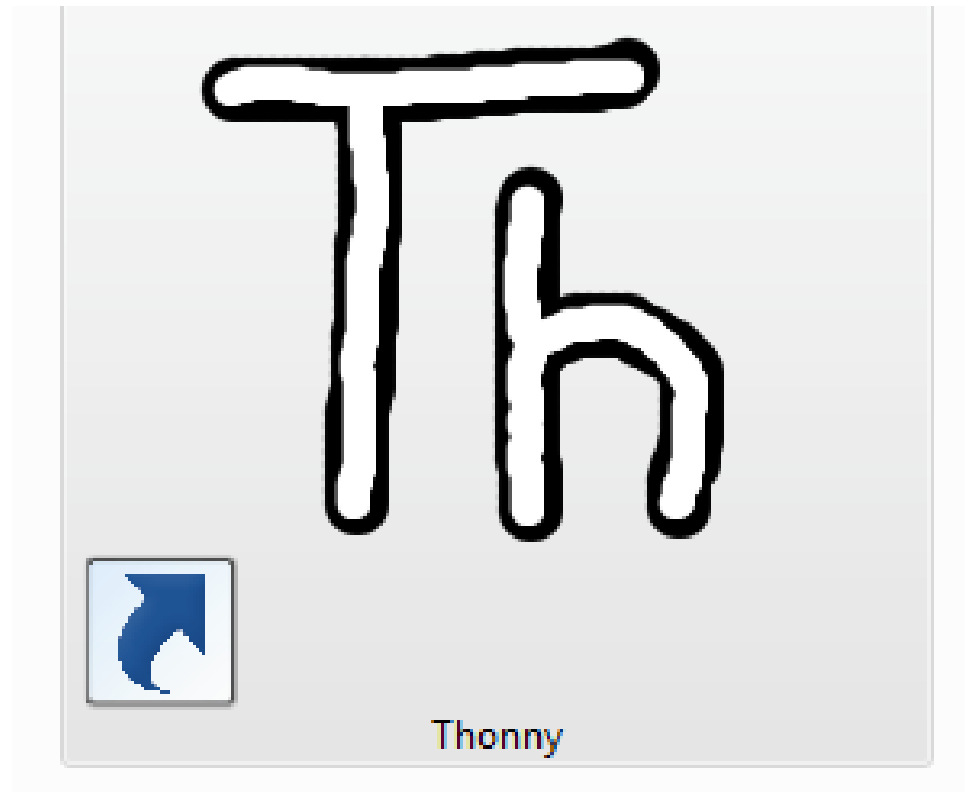
```
a=int(input())
```

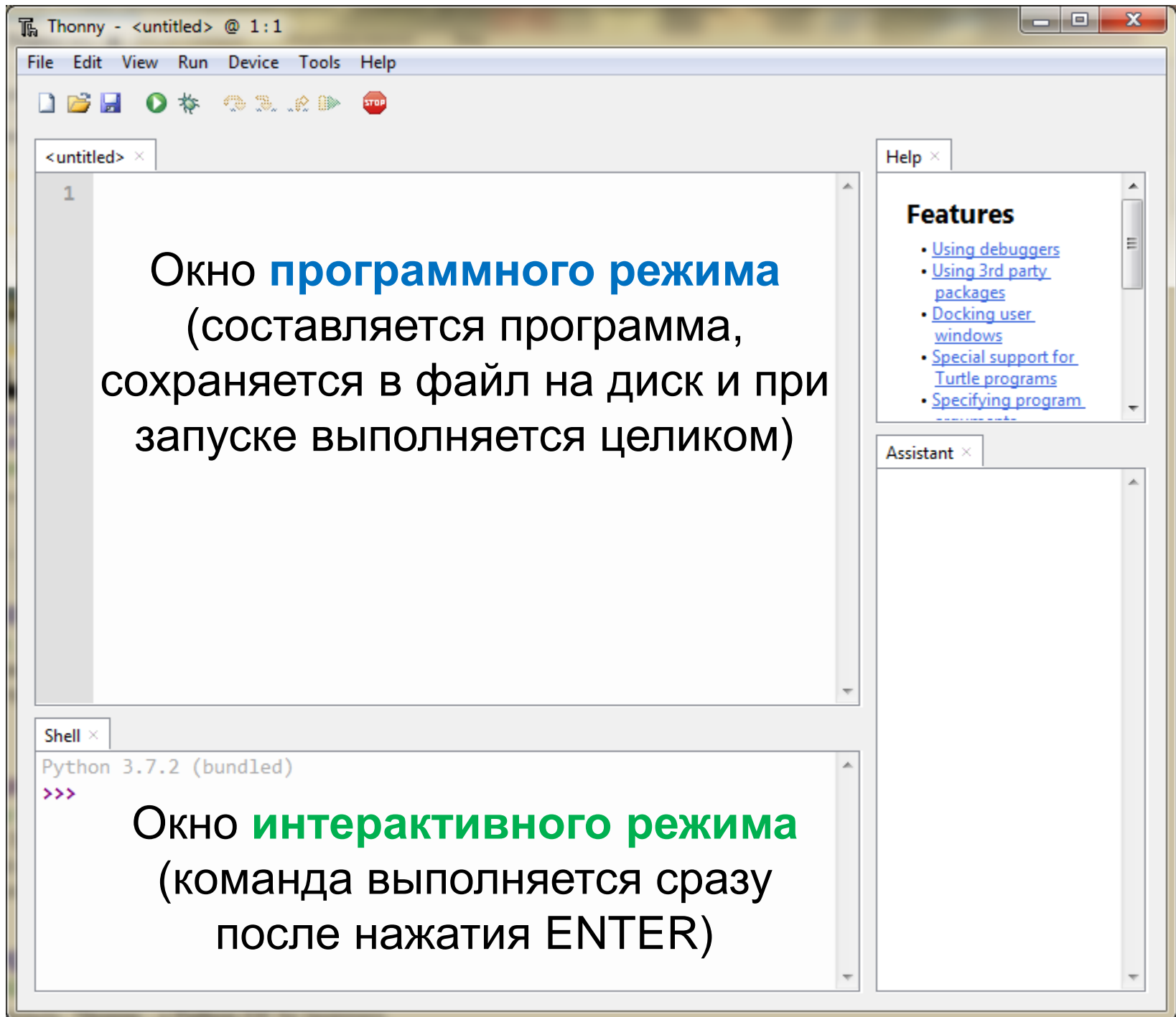
Ввод **целого** числа

```
a=float(input())
```

Ввод **любого** числа
(с плавающей точкой)

Запуск интегрированной среды разработки (для новичков)





Найти произведение 3 любых целых чисел, введенных с клавиатуры

ВВОД a

ВВОД b

ВВОД c

$x = a * b * c$

ВЫВОД x

Алгоритм

Найти произведение 3 любых целых чисел, введенных с клавиатуры

ВВОД a
ВВОД b
ВВОД c
 $x = a * b * c$
ВЫВОД x

Алгоритм

```
1 a=int(input())  
2 b=int(input())  
3 c=int(input())  
4 x=a*b*c  
5 print(x)
```

Программа

Запускаем программу (F5)

1.py ×

```
1 a=int(input())  
2 b=int(input())  
3 c=int(input())  
4 x=a*b*c  
5 print(x)
```

Shell ×

Запускаем программу (F5)

1.py ×

```
1 a=int(input())  
2 b=int(input())  
3 c=int(input())  
4 x=a*b*c  
5 print(x)
```

Shell ×

```
>>> %Run 1.py
```

```
3
```

```
>>>
```

Вводим целые значения и
нажимаем ENTER

Запускаем программу (F5)

1.py ×

```
1 a=int(input())
2 b=int(input())
3 c=int(input())
4 x=a*b*c
5 print(x)
```

Shell ×

```
>>> %Run 1.py
```

```
3
```

```
4
```

```
>>>
```

Вводим целые значения и
нажимаем ENTER

Запускаем программу (F5)

1.py ×

```
1 a=int(input())  
2 b=int(input())  
3 c=int(input())  
4 x=a*b*c  
5 print(x)
```

Shell ×

```
>>> %Run 1.py
```

```
3  
4  
5
```

```
>>>
```

Вводим целые значения и
нажимаем ENTER

Запускаем программу (F5)

1.py ×

```
1 a=int(input())
2 b=int(input())
3 c=int(input())
4 x=a*b*c
5 print(x)
```

Shell ×

```
>>> %Run 1.py
```

```
3
```

```
4
```

```
5
```

```
60
```

```
>>>
```

Получаем результат

Работа 3.1. Программирование линейных алгоритмов

Задание

Для каждой вычислительной задачи составить программу, содержащую операторы ввода, вывода, присваивания.

Уровень 1

1. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .
2. Вычислить расстояние между двумя точками с данными координатами на плоскости (x_1, y_1) и (x_2, y_2) .
3. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
4. Три сопротивления R_1, R_2, R_3 соединены параллельно. Найти сопротивление всей цепи.
5. Найти сумму членов арифметической прогрессии, если известны ее первый член, разность и число членов прогрессии.
6. Вычислить корни квадратного уравнения $ax^2 + bx + c = 0$ с заданными коэффициентами a, b и c (предполагается, что $a \neq 0$ и что дискриминант уравнения неотрицателен).
7. Найти площадь равнобедренной трапеции с основаниями a и b и углом α при большем основании a .

Разберем задачу №1

1. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .

Разберем задачу №1

1. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .

Формализация задачи

Входные данные – ?

Выходные данные – ?

Формулы для вычисления:

$L = ?$

$S = ?$

Разберем задачу №1

1. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .

Формализация задачи

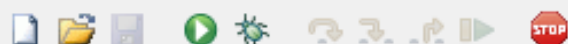
Входные данные – R

Выходные данные – L , S

Формулы для вычисления:

$$L = 2\pi R$$

$$S = \pi R^2$$



000.py ×

```
1 import math
2 r=int(input('введите радиус:'))
3 l=2*math.pi*r
4 s=math.pi*r**2
5 print('длина=',l)
6 print ('площадь=',s)
```

Shell ×

```
>>> %Run 000.py
```

```
введите радиус:2
длина= 12.566370614359172
площадь= 12.566370614359172
```

```
>>> %Run 000.py
```

```
введите радиус:4
длина= 25.132741228718345
площадь= 50.26548245743669
```

```
>>> |
```

Задание на дом:

**составить программу
для решения любой
задачи из учебника
стр.231**