

|                    |  |
|--------------------|--|
| <b>tan(x)</b>      | Тангенс угла, задаваемого в радианах                           |
| <b>asin(x)</b>     | Арксинус, возвращает значение в радианах                       |
| <b>acos(x)</b>     | Арккосинус, возвращает значение в радианах                     |
| <b>atan(x)</b>     | Арктангенс, возвращает значение в радианах                     |
| <b>atan2(y, x)</b> | Полярный угол (в радианах) точки с координатами (x, y).        |
| <b>hypot(a, b)</b> | Длина гипотенузы прямоугольного треугольника с катетами a и b. |
| <b>degrees(x)</b>  | Преобразует угол, заданный в радианах, в градусы.              |
| <b>radians(x)</b>  | Преобразует угол, заданный в градусах, в радианы.              |
| <b>pi</b>          | Константа $\pi$  |

Некоторые из перечисленных функций (int, round, abs) являются стандартными и не требуют подключения модуля math для использования.

### **Оборудование и материалы.**

Персональный компьютер, среда разработки Python.

### **Указания по технике безопасности:**

Соответствуют технике безопасности по работе с компьютерной техникой.

### **Задания**

Для начала вернемся к стандартной библиотеке и рассмотрим некоторые особенности и модули.

#### ***Математические функции***

Python имеет множество математических функций в стандартной библиотеке math. Просто введите import math, чтобы получить к ним доступ из своих программ.

Она содержит такие константы, как pi и e:

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.e
2.718281828459045
```

В основном код состоит из функций, рассмотрим наиболее полезные из них.

Функция fabs() возвращает абсолютное значение своего аргумента:

```
>>> math.fabs(98.6)
98.6
>>> math.fabs(-271.1)
271.1
```

Получаем округление вниз (floor()) и вверх (ceil()) некоторого числа:

```
>>> math.floor(98.6)
98
>>> math.floor(-271.1)
-272
>>> math.ceil(98.6)
99
>>> math.ceil(-271.1)
-271
```

Вычисляем факториал (в математике это выглядит как n!) с помощью функции factorial():

```
>>> math.factorial(0)
1
>>> math.factorial(1)
1
>>> math.factorial(2)
2
>>> math.factorial(3)
6
>>> math.factorial(10)
3628800
```

Получаем натуральный логарифм аргумента с помощью функции `log()`:

```
>>> math.log(1.0)
0.0
>>> math.log(math.e)
1.0
```

Если вы хотите задать другое основание логарифма, передайте его как второй аргумент:

```
>>> math.log(8, 2)
3.0
```

Функция `pow()` возвращает противоположный результат, возводя число в степень:

```
>>> math.pow(2, 3)
8.0
```

Python имеет также встроенный оператор экспоненты `**`, делающий то же самое, но он не преобразует автоматически результат к числу с плавающей точкой, если основание и степень были целыми числами:

```
>>> 2**3
8
>>> 2.0**3
8.0
```

Получаем квадратный корень с помощью функции `sqrt()`:

```
>>> math.sqrt(100.0)
10.0
```

Если вы попытаетесь задать отрицательный аргумент, то получите:

```
>>> math.sqrt(-100.0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: math domain error
```

Здесь имеются также обычные тригонометрические функции, я просто приведу их названия: `sin()`, `cos()`, `tan()`, `asin()`, `acos()`, `atan()` и `atan2()`. Если вы помните теорему Пифагора (или хотя бы можете произнести ее название быстро три раза, не начав

плываться), библиотека `math` предоставит вам также функцию `hypot()`, которая рассчитывает гипотенузу на основании длины двух катетов:

```
>>> x = 3.0
>>> y = 4.0
>>> math.hypot(x, y)
5.0
```

Если вы не доверяете таким функциям, можете сделать все самостоятельно:

```
>>> math.sqrt(x*x + y*y)
5.0
>>> math.sqrt(x**2 + y**2)
5.0
```

Последний набор функций преобразует угловые координаты:

```
>>> math.radians(180.0)
3.141592653589793
>>> math.degrees(math.pi)
180.0
```

### Работа с комплексными числами

Комплексные числа полностью поддерживаются основой языка Python в традиционной нотации с мнимой и действительной частями:

```
>>> # действительное число
... 5
5
>>> # мнимое число
... 8j
8j
>>> # мнимое число
... 3 + 2j
(3+2j)
```

Поскольку мнимое число  $i$  (в Python оно записывается как `1j`) определено как квадратный корень из  $-1$ , мы можем выполнить следующее:

```
>>> 1j * 1j
(-1+0j)
>>> (7 + 1j) * 1j
(-1+7j)
```

Некоторые математические функции для комплексных чисел содержатся в стандартном модуле `cmath`.

### Рассчитываем точное значение чисел с плавающей точкой с помощью `decimal`

Числа с плавающей точкой в вычислительной технике не похожи на настоящие числа, которые мы изучали в школе. Из-за того что компьютеры разрабатывались для



Получим наибольший общий делитель для двух чисел с помощью функции gcd:

```
>>> import fractions
>>> fractions.gcd(24, 16)
8
```

### **Контрольные вопросы**

1. Каково назначение модуля math? Его основные функции?
2. Функции округления, особенности их использования.
3. Константы модуля math.
4. Как осуществляется работа с комплексными числами в Python?
5. Использование модуля decimal.
6. Использование модуля fractions.

### **Список литературы, рекомендуемый к использованию по данной теме:**

1. Орлов, С. А. Программная инженерия. Технологии разработки программного обеспечения : учебник / С.А. Орлов. - 5-е изд., обновл. и доп. - СПб. : Питер, 2017. - 640 с.
2. Гагарина, Л. Г. Современные проблемы информатики и вычислительной техники : [учеб. пособие] / Л.Г. Гагарина, А.А. Петров. - М. : Форум, 2016. - 368 с.
3. Шкляр, М. Ф. Основы научных исследований : учеб. пособие / М.Ф. Шкляр. - 6-е изд. - М. : Дашков и Ко, 2016. - 208 с.
4. Михеева, Е. В. Информационные технологии в профессиональной деятельности : учеб. пособие / Е.В. Михеева. - 14-е изд., стер. - М. : Академия, 2016. - 384 с.
5. Любанович Билл. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.

## **Лабораторная работа 18. Модуль для научных расчетов NumPy. Работа с массивами.**

### **Цель работы:**

Изучить особенности создания массивов в NumPy. Создание массива с помощью функций zeros(), ones() и random(). Изменение формы массива с помощью метода reshape(). Получение элемента с помощью конструкции []. Математика массивов. Линейная алгебра.

### **Компетенции:**

| Код   | Формулировка:   |
|-------|---|
| ОК-5  | быть готовым работать с информацией в различных формах, использовать для ее получения, обработки, передачи, хранения и защиты современные компьютерные технологии   |
| ОПК-5 | способностью использовать основные приемы обработки и представления экспериментальных данных  |
| ОПК-6 | способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий |