

# Программирование на языке Java

- 10. Типы с плавающей точкой
- 11. Методы класса **Методы класса**  
Math

# Программирование на языке Java

**Тема 10. Типы с  
плавающей точкой**

# Типы с плавающей точкой – 1

---

**Числа с плавающей точкой**, называемые **действительными**, используются при вычислениях, которые требуют получения результата с точностью до определенного десятичного знака.

**Пример.** Вычисление квадратного корня, трансцендентных функций (`sin()` , `cos()` , ...).

В Java 2 типа с плавающей точкой: `float` и `double` (числа одинарной и двойной точности).

## Типы с плавающей точной – 2

---

Тип	Размер	Min	Max
<code>double</code> (двойная точность)	64 бит	4.9e-324	1.8e308
<code>float</code> (одинарная точность)	32 бита	1.4e-045	3.4e038

## Типы с плавающей точной – 3

---

Тип `float` используется, когда требуется дробная часть без **особой точности**, например для представления денежных сумм в рублях и копейках.

Применение типа `double` наиболее рационально, когда требуется сохранение точности множества последовательных вычислений или манипулирование большими числами.

Все трансцендентные математические функции (`sin()`, `cos()`, `sqrt()`, ...) возвращают значения типа `double`.

# Задача

**Задача.** Вычислить площадь круга заданного радиуса.

Считываем с клавиатуры

```
double pi, a, r;  
r = in.nextDouble();  
pi = 3.1415926;  
a = pi * r * r;  
System.out.println(a);
```

Приблизительное  
значение

Вычисление площади  
круга

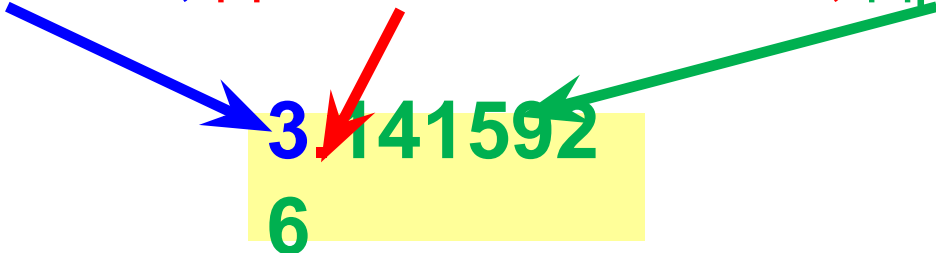
# Константы с плавающей точкой – 1

---

Числа с плавающей точкой представляют десятичные значения с дробной частью.

**Стандартная форма записи десятичного числа** состоит из:

целого числа; десятичной точки; дробной части.



3.1415926

# Константы с плавающей точкой – 2

---

Научная форма записи десятичного числа  
СОСТОИТ ИЗ:

числа; символа E, суффикса, указывающего  
степенную функцию числа 10



314159e-5

The diagram illustrates the components of scientific notation. A blue arrow points from the word 'числа' (number) to the digits '314159'. A red arrow points from the word 'символа E' (symbol E) to the 'e' in 'e-5'. A green arrow points from the phrase 'суффикса, указывающего степенную функцию числа 10' (suffix indicating the power function of the number 10) to the '-5'.



## Константы с плавающей точкой – 3

---

**Задача.** Записать в стандартной форме

$$1.44\text{e-}6 = 0.00000144$$

$$0.832\text{e}8 = 83200000$$

$$0.000034\text{e}7 = 340$$

$$0.00524\text{e-}1 = 0.000524$$

## Константы с плавающей точкой – 3

---

По умолчанию в **Java** константам с плавающей точкой присвоен тип **double**.

Для указания константы типа **float**, к ней нужно дописать символ **F** или **f**.

```
float x;  
x = 23.48f;
```

# Программирование на языке Java

**Тема 11. Методы класса  
Math**

# Класс `Math`

---

Разработчику на `Java` доступно множество готовых (или библиотечных) классов и методов, полезных для использования в собственных программах.

Наличие библиотечных решений позволяет изящно решать множество типовых задач.

Класс `Math` содержит все функции с плавающей точкой, которые используются в геометрии и тригонометрии, а также некоторые методы общего назначения.

# Константы класса Math

~3,14

2 константы типа `double`:

~2,72

`Math.PI` — число  $\pi$  с точностью в 15 десятичных знаков.

`Math.E` — основание натурального логарифма с точностью в 15 десятичных знаков.

```
System.out.println(Math.PI);  
System.out.println(Math.E);
```

```
3.141592653589793  
2.718281828459045
```

# Прямые трансцендентные функции

Метод	Описание
<code>double sin(double arg)</code>	Возвращает синус угла <code>arg</code> , переданного в радианах
<code>double cos(double arg)</code>	Возвращает косинус угла <code>arg</code> , переданного в радианах
<code>double tan(double arg)</code>	Возвращает тангенс угла <code>arg</code> , переданного в радианах

# Прямые трансцендентные функции. Пример

---

```
System.out.println(Math.sin(Math.PI/2));  
System.out.println(Math.cos(Math.PI/2));  
System.out.println(Math.tan(Math.PI/4));
```

1.0

6.123233995736766E-17

0.99999999999999999999

# Обратные трансцендентные функции

Метод	Описание
<code>double asin(double arg)</code>	Возвращает угол, синус которого равен <code>arg</code> .
<code>double acos(double arg)</code>	Возвращает угол, косинус которого равен <code>arg</code> .
<code>double atan(double arg)</code>	Возвращает угол, тангенс которого равен <code>arg</code> .
<code>double atan2(double x, double y)</code>	Возвращает угол, тангенс которого равен $x/y$ .



# Обратные трансцендентные функции. Пример

---

```
System.out.println(Math.asin(1)*2) ;  
System.out.println(Math.acos(1)) ;  
System.out.println(1) ;  
System.out.println(Math.atan2(1,1)*4) ;
```

```
3.141592653589793
```

```
0.0
```

```
1
```

```
3.141592653589793
```

# Гиперболические функции

Метод	Описание
<code>double sinh(double arg)</code>	Возвращает гиперболический синус угла <code>arg</code> , переданного в радианах.
<code>double cosh(double arg)</code>	Возвращает гиперболический косинус угла <code>arg</code> , переданного в радианах.
<code>double tanh(double arg)</code>	Возвращает гиперболический тангенс угла <code>arg</code> , переданного в радианах.

# Экспоненциальные функции

Метод	Описание
<code>double exp(double arg)</code>	Возвращает экспоненту <code>arg</code> .
<code>double log(double arg)</code>	Возвращает натуральный логарифм <code>arg</code> .
<code>double log10(double arg)</code>	Возвращает логарифм по основанию 10 от <code>arg</code> .
<code>double pow(double y, double x)</code>	Возвращает <code>y</code> в степени <code>x</code> .
<code>double sqrt(double arg)</code>	Возвращает квадратный корень из <code>arg</code> .

# Экспоненциальные функции. Пример

---

```
System.out.println(Math.exp(1));  
System.out.println(Math.exp(2));  
System.out.println(Math.log(1));  
System.out.println(Math.log(Math.E));  
System.out.println(Math.log10(1000));  
System.out.println(Math.pow(2, 3));  
System.out.println(Math.sqrt(25));
```

2.7182818284590455

7.38905609893065

0.0

1.0

3.0

8.0

5.0

# Функции округления – 1

Метод	Описание
<code>int abs(int arg)</code>	Возвращает абсолютное значение <code>arg</code> .
<code>long abs(long arg)</code>	Возвращает абсолютное значение <code>arg</code> .
<code>float abs(float arg)</code>	Возвращает абсолютное значение <code>arg</code> .
<code>double abs(double arg)</code>	Возвращает абсолютное значение <code>arg</code> .

# Функции округления – 1. Пример

---

```
System.out.println(Math.abs(5));  
System.out.println(Math.abs(-5));  
System.out.println(Math.abs(10.3));  
System.out.println(Math.abs(-10.3));
```

```
5  
5  
10.3  
10.3
```

## Функции округления – 2

Метод	Описание
<code>double ceil(double arg)</code>	Возвращает наименьшее целое число, которое больше <code>arg</code> .
<code>double floor(double arg)</code>	Возвращает наибольшее целое число, которое меньше или равно <code>arg</code> .
<code>int round(float arg)</code>	Возвращает <code>arg</code> , округленное вверх до ближайшего <code>int</code> .
<code>long round(double arg)</code>	Возвращает <code>arg</code> , округленное вверх до ближайшего <code>long</code> .

## Функции округления – 2. Пример

потолок

```
System.out.println(Math.ceil(5.4));  
System.out.println(Math.floor(5.4));  
System.out.println(Math.round(5.4));  
System.out.println(Math.round(5.6));  
System.out.println(Math.round(5.5));
```

6.0

5.0

5

6

6

пол



## Функции округления – 3

Метод	Описание
<code>int max(int x, int y)</code>	Возвращает большее из двух чисел <b>x</b> и <b>y</b> .
<code>long max(long x, long y)</code>	Возвращает большее из двух чисел <b>x</b> и <b>y</b> .
<code>float max(float x, float y)</code>	Возвращает большее из двух чисел <b>x</b> и <b>y</b> .
<code>double max(double x, double y)</code>	Возвращает большее из двух чисел <b>x</b> и <b>y</b> .

## Функции округления – 3

Метод	Описание
<code>int min(int x, int y)</code>	Возвращает меньшее из двух чисел <b>x</b> и <b>y</b> .
<code>long min(long x, long y)</code>	Возвращает меньшее из двух чисел <b>x</b> и <b>y</b> .
<code>float min(float x, float y)</code>	Возвращает меньшее из двух чисел <b>x</b> и <b>y</b> .
<code>double min(double x, double y)</code>	Возвращает меньшее из двух чисел <b>x</b> и <b>y</b> .

## Функции округления – 3. Пример

---

```
System.out.println(Math.max(2,4));  
System.out.println(Math.min(2,4));  
System.out.println(Math.max(10.3,4.5));  
System.out.println(Math.min(10.3,4.5));
```

4

2

10.3

4.5

# Прочие функции

Метод	Описание
<code>double toDegrees (double angle)</code>	Преобразует радианы в градусы. Переданный в <code>angle</code> угол должен быть указан в радианах. Возвращается результат в градусах.
<code>double toRadians (double angle)</code>	Преобразует градусы в радианы. Переданный в <code>angle</code> угол должен быть указан в градусах. Возвращается результат в радианах.

## Прочие функции. Пример

---

```
System.out.println(Math.toDegrees(Math.PI));  
System.out.println(Math.toDegrees(Math.PI/4))  
;  
System.out.println(Math.toRadians(180));  
System.out.println(Math.toRadians(90));
```

180.0

45.0

3.141592653589793

1.5707963267948966

# Псевдослучайные числа

---

Метод `Math.random()` возвращает псевдослучайное вещественное число из промежутка  $[0;1)$ .

```
System.out.println(Math.random());  
System.out.println(Math.random());  
System.out.println(Math.random());
```

```
0.8701659383706429  
0.5194884184661862  
0.3324845299964946
```

# Целые числа в заданном интервале – 1

---

Целые числа в интервале  $[0, N-1]$ :

```
(int) (Math.random() * N) ;
```

Примеры:

```
x = (int) (Math.random () * 100) ; // [0,99]
```

```
x = (int) (Math.random () * z) ; // [0,z-1]
```

Целые числа в интервале  $[a, b]$ :

```
x = (int) (Math.random () * z) + a; // [a,z-1+a]
```

```
x = (int) (Math.random () * (b - a + 1)) + a;
```

```
// [a,b]
```

# Целые числа в заданном интервале – 1

---

**Задача.** Получить случайное число в интервале от -10 до 10.

```
int x = random () * 21 - 10;
```



# Методы класса Math. Задача – 1

---

```
System.out.println(Math.abs(-2.33));  
System.out.println(Math.round(Math.PI));  
System.out.println(Math.round(9.5));  
System.out.println(Math.round(9.5-0.001));  
;  
System.out.println(Math.ceil(9.4));  
double c = Math.sqrt(3*3 + 4*4);  
System.out.println(c);  
double s1 = Math.cos(Math.toRadians(60));  
System.out.println(s1);
```

2.33

3

10

9

10.0

5.0

0.5

## Методы класса Math. Задача – 2

---

Записать в стандартной форме записи числа

$$-12.3\text{E}+2 = -1230$$

$$-0.8\text{E}-6 = -0.0000008$$

$$1\text{E}3 = 1000$$

$$+1\text{E}-6 = 0.000001$$

## Методы класса Math. Задача – 3

---

Какие круглые скобки можно убрать, не изменив порядка вычисления выражений

$(a+b)/c$

$a+b/c$

$a/(b*c)$

$x1/x2*y$

$\text{Math.sqrt}(p)*q/r$

$a-b-c-d-e$

$(a-b)-(c-d)-e$

## Методы класса Math. Задача – 4

---

Записать следующие выражения на Java

$x^5$

```
Math.pow(x, 5)
```

$\cos^8 x^4$

```
Math.pow(Math.cos(Math.pow(x, 4)), 8)
```

$\log_{10}(x/5)$

```
Math.log10(x/5)
```

$|x^{-3}|$

```
Math.abs(Math.pow(x, -3))
```

$2^{x+1}$

```
Math.pow(2, x+1)
```

$\sin 8^\circ$

```
Math.sin(Math.toRadians(8))
```

# Методы класса Math. Задача – 5

---

Определить типы выражений

```
double x, y, z;
```

```
int i, j, k;
```

```
x+y*i;
```

```
i+j-k;
```

```
i/j+x;
```

```
i*x+j*y;
```

double

int

double

double