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 точной – 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 точной: **`float`** и **`double`** (числа одинарной и двойной точности).

Типы с плавающей точной – 2

Тип	Размер	Min	Max
`double` (двойная точность)	64 бит	4.9e-324	1.8e308
`float`(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.

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

Считываем с клавиатуры

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.

3.141592 6

Константы с плавающей точкой – 2

Научная форма записи десятичного числа состоит из:

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

314159e–5

Константы с плавающей точкой – 3

Задача. Записать в стандартной форме

1.44e-6 = 0.00000144

0.832e8= 83200000

0.000034e7= 340

0.00524e-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

~2,72

2 константы типа **`double`**:

`Math.PI` – число π 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
`double sin(double arg)`	Возвращает синус угла **`arg`**, переданного в радианах
`double cos(double arg)`	Возвращает косинус угла **`arg`**, переданного в радианах
`double tan(double arg)`	Возвращает тангенс угла **`arg`**, 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
0.9999999999999999
```

Обратные трансцендентные функции

Метод	Описание
`double asin(double arg)`	Возвращает угол, синус которого равен **`arg`**.
`double acos(double arg)`	Возвращает угол, косинус которого равен **`arg`**.
`double atan(double arg)`	Возвращает угол, тангенс которого равен **`arg`**.
`double atan2(double x, double y)`	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
`double sinh(double arg)`	Возвращает гиперболический синус угла **`arg`**, переданного в радианах.
`double cosh(double arg)`	Возвращает гиперболический косинус угла **`arg`**, переданного в радианах.
`double tanh(double arg)`	Возвращает гиперболический тангенс угла **`arg`**, переданного в радианах.

Экспоненциальные функции

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

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
`int abs(int arg)`	Возвращает абсолютное значение **`arg`**.
`long abs(long arg)`	Возвращает абсолютное значение **`arg`**.
`float abs(float arg)`	Возвращает абсолютное значение **`arg`**.
`double abs(double arg)`	Возвращает абсолютное значение **`arg`**.

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
`double ceil(double arg)`	Возвращает наименьшее целое число, которое больше **`arg`**.
`double floor(double arg)`	Возвращает наибольшее целое число, которое меньше или равно **`arg`**.
`int round(float arg)`	Возвращает **`arg`**, округленное вверх до ближайшего **`int`**.
`long round(double arg)`	Возвращает **`arg`**, округленное вверх до ближайшего **`long`**.

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
`int max(int x, int y)`	Возвращает большее из двух чисел **x** и **y**.
`long max(long x, long y)`	Возвращает большее из двух чисел **x** и **y**.
`float max(float x, float y)`	Возвращает большее из двух чисел **x** и **y**.
`double max(double x, double y)`	Возвращает большее из двух чисел **x** и **y**.

Функции округления – 3

Метод	Описание
`int min(int x, int y)`	Возвращает меньшее из двух чисел **x** и **y**.
`long min(long x, long y)`	Возвращает меньшее из двух чисел **x** и **y**.
`float min(float x, float y)`	Возвращает меньшее из двух чисел **x** и **y**.
`double min(double x, double y)`	Возвращает меньшее из двух чисел **x** и **y**.

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
`double toDegrees(double angle)`	Преобразует радианы в градусы. Переданный в **`angle`** угол должен быть указан в радианах. Возвращается результат в градусах.
`double toRadians(double angle)`	Преобразует градусы в радианы. Переданный в **`angle`** 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

-12.3E+2 **= -1230**

-0.8E-6 **= -0.0000008**

1E3 **= 1000**

+1E-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

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
)
```

$\sin 8°$

```
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