

## Análise e Desenvolvimento de Sistemas – ADS Linguagem de Programação – LP

### A Classe MATH em Java

Prof. Cristóvão Cunha

#### Assunto: pacote `java.lang.Math`

*Esta apostila visa explicar e exemplificar as constantes e os métodos que estão no pacote `java.lang.Math` (Classe Math do Java), ajudando a aprender e entender como funcionam e como utilizar.*

### A Classe Java Math

A classe `java Math` fornece mais cálculos matemáticos avançados, do que o que os operadores matemáticos básicos Java fornecem. A classe `Math` contém métodos para encontrar o máximo ou mínimo de dois valores, arredondar valores, funções logarítmicas, raiz quadrada e funções trigonométricas (`sin`, `cos`, `tan`, etc).

A classe `Math` está localizado no pacote `java.lang`, e não no pacote `java.Math`. Assim, o nome de classe totalmente qualificado da classe `Math` é `java.lang.Math`.

Uma vez que muitas das funções da classe `Math` são independentes uma da outra, tentaremos explicar o maior número de métodos, exemplificando sua utilização.

### Funções básicas de matemática

O `java.lang.Math` contém um conjunto de funções matemáticas básicas para obter o valor absoluto, maior e menor de dois valores, o arredondamento dos valores, valores aleatórios, etc. Estas funções matemáticas básicas da classe `Java Math` serão abordadas a seguir.

#### **Math.abs ()**

O `Math.abs ()` devolve o valor absoluto do parâmetro passado para ele. O valor absoluto é o valor positivo do parâmetro. Se o valor do parâmetro é negativo, o sinal negativo é removido e o valor positivo que corresponde ao valor negativo sem sinal é devolvido. Aqui estão dois exemplos de métodos `Math.abs ()`:

```
int abs1 = Math.abs (10); // abs1 = 10
```

```
int abs2 = Math.abs (-20); // abs2 = 20
```

O valor absoluto de 10 é 10 e o valor absoluto de -20 a 20.

O método `Math.abs()` é sobrecarregado em 4 versões:

```
Math.abs (int)  
Math.abs (long)  
Math.abs (float)  
Math.abs (double)
```

O retorno destes métodos depende do tipo do parâmetro transmitido para o ele, ou seja, se a entrada for um int o retorno será um int, se a entrada for um long, o retorno será um long e assim por diante.

*Referência: MathAbs.java*

## **Math.ceil ()**

O método `Math.ceil()` arredonda um valor de ponto flutuante até o valor inteiro mais próximo.

O retorno do valor arredondado é um double. Exemplo:

```
double ceil = Math.ceil (7.343); // ceil = 8.0
```

Depois de executar este código Java a variável `ceil` irá conter o valor 8.0.

*Referência: MathCeil.java*

## **Math.floor ()**

O método `Math.floor()` arredonda um valor ponto flutuante para o maior valor inteiro mais próximo que não seja maior que o número. O valor arredondado é retornado como um double. Exemplo:

```
double floor = Math.floor (7.343); // floor = 7.0
```

Depois de executar este código Java a variável `floor` irá conter o valor 7.0.

*Referência: MathFloor.java*

## **Math.min ()**

O método `Math.min()` retorna o menor dos dois valores que são passados como parâmetros. Exemplo:

```
int min = Math.min (10, 20);
```

Depois de executar este código a variável `min` irá conter o valor 10.

Assim com o método `Math.abs()`, o método `Math.min()` é sobrecarregado em 4 versões:

```
Math.min (int, int)
Math.min (long, long)
Math.min (float, float)
Math.min (double, double)
```

O retorno destes métodos depende do tipo do parâmetro transmitido para o ele. Caso sejam de tipos diferentes, prevalece o maior dos tipos.

*Referência: MathMin.java*

## Math.max ()

O método `Math.max()` retorna o maior dos dois valores que são passados como parâmetros. Exemplo:

```
int max = Math.max (10, 20);
```

Depois de executar este código a variável `max` irá conter o valor 20.

Assim com os métodos `Math.abs()` e `Math.min()`, o método `Math.max()` é sobrecarregado em 4 versões:

```
Math.max (int, int)
Math.max (long, long)
Math.max (float, float)
Math.max (double, double)
```

O retorno destes métodos depende do tipo do parâmetro transmitido para o ele. Caso sejam de tipos diferentes, prevalece o maior dos tipos.

*Referência: MathMax.java*

## Math.round ()

O método `Math.round()` arredonda um `float` ou `double` para o número inteiro mais próximo usando as regras normais de arredondamento da matemática (para cima ou para baixo). Exemplo:

```
int roundedDown = Math.round (23.445F);
long roundedUp   = Math.round (23.545D);
```

Depois de executar essas duas declarações do Java, a variável `roundedDown` irá conter o valor 23, e a variável `roundedUp` irá conter o valor 24.

O método `Math.round()` é sobrecarregado em 2 versões:

```
Math.round (float)
Math.round (double)
```

O retorno destes métodos depende do tipo do parâmetro transmitido para o ele. Caso seja passado um float o retorno é um int e caso seja passado um double o retorno será um long.

*Referência: MathRound.java*

## Math.random ()

O método `Math.random()` retorna um número aleatório de ponto flutuante entre 0 (incluso) e 1 (excluso). É claro que o número não é totalmente aleatória, mas o resultado de algum cálculo que é suposto para torná-lo tão imprevisível quanto possível. Exemplo:

```
double random = Math.random ();
```

Para obter um valor aleatório entre 0 (incluso) e 100 (excluso), por exemplo, multiplicar o valor retornado por `Math.random()` com o número máximo (por exemplo, 100). Exemplo:

```
double random = Math.random () * 100D;
```

Se você precisar de um valor inteiro, pode utilizar os métodos da própria classe `Math`, como o `round()`, o `floor()` ou o `ceil()`, ou ainda, se utilizar do `Cast` para o tipo desejado.

```
int random = (int) (Math.random () * 100);
```

*Referência: MathRandom.java*

## Funções Matemáticas exponenciais e logarítmicas

A classe `Java Math`, também, contém um conjunto de funções para cálculos exponenciais e logarítmicos. Algumas dessas funções são descritas a seguir.

### Math.exp ()

O método `Math.exp()` retorna  $e$  (número de Euler) elevado à potência do valor fornecido como parâmetro. Exemplo:

```
double exp1 = Math.exp (1);
System.out.println ("exp1 = " + exp1);

double exp2 = Math.exp (2);
System.out.println ("exp2 = " + exp2);
```

Quando este código Java de matemática é executado ele irá imprimir esta saída:

```
exp1 = 2.718281828459045  
exp2 = 7.38905609893065
```

*Referência: MathExp.java*

## Math.log ()

O método `Math.log()` fornece o logaritmo do parâmetro determinado. A base para o logaritmo é *e* (número de Euler). Assim, `Math.log()` fornece a função inversa da `Math.exp()`. Exemplo:

```
double log1 = Math.log (1);  
System.out.println ("log1 = " + log1);  
  
double log10 = Math.log (10);  
System.out.println ("log10 = " + log10);
```

A saída é:

```
log1 = 0.0  
log10 = 2.302585092994046
```

*Referência: MathLog.java*

## Math.pow ()

O método `Math.pow()` tem dois parâmetros. O método retorna o valor do primeiro parâmetro elevado à potência do segundo parâmetro. Exemplo:

```
double pow1 = Math.pow (2,2);  
System.out.println ( "pow1 = " + pow1);  
  
double pow2 = Math.pow (2,8);  
System.out.println ( "pow2 = " + pow2);
```

A saída é:

```
pow1 = 4.0  
pow2 = 256.0
```

Em outras palavras, o exemplo `Math.pow()`, calcula os valores de  $2^2$  e  $2^8$ , que são 4 e 256.

*Referência: MathPow.java*

## Math.sqrt ()

O método `Math.sqrt()` calcula a raiz quadrada do parâmetro que lhe é dado. Exemplo:

```
double sqrt1 = Math.sqrt (4);  
System.out.println ( "sqrt1 = " + sqrt1);  
  
double sqrt2 = Math.sqrt (9);  
System.out.println ( "sqrt2 = " + sqrt2);
```

A saída é:

```
sqrt1 = 2.0  
sqrt2 = 3.0
```

*Referência: MathSqrt.java*

## Funções trigonométricas matemática

A classe Java Math contém um conjunto de funções trigonométricas. Essas funções podem calcular os valores utilizados na trigonometria, como seno, cosseno, tangente etc. Serão apresentadas as funções de trigonometria mais utilizadas a seguir.

### Math.PI

A constante `Math.PI` é um valor do tipo `double`, que é muito próximo do valor de PI (da definição matemática do PI). Muitas vezes você vai precisar do `Math.PI` ao fazer cálculos trigonométricos. Exemplo:

```
System.out.println ( "PI = " + Math.PI );
```

*Referência: MathPI.java*

### Math.sin ()

O método `Math.sin()` calcula o valor do seno de algum valor do ângulo em radianos. Exemplo:

```
double sin = Math.sin (Math.PI);  
System.out.println ( "sin = " + sin );
```

*Referência: MathSin.java*

### Math.cos ()

O método `Math.cos()` calcula o valor do cosseno de algum valor do ângulo em radianos. Exemplo:

```
double cos = Math.cos (Math.PI);  
System.out.println ( "cos = " + cos );
```

*Referência: MathCos.java*

## Math.tan ()

O método `Math.tan()` calcula o valor tangente de algum valor do ângulo em radianos. Exemplo:

```
double tan = Math.tan (Math.PI);  
System.out.println ( "tan = " + tan );
```

*Referência: MathTan.java*

## Math.asin ()

O método `Math.asin()` calcula o valor do arco seno de um valor entre 1 e -1. Exemplo:

```
double asin = Math.asin (1.0);  
System.out.println ( "asin = " + asin );
```

*Referência: MathAsin.java*

## Math.acos ()

O método `Math.acos()` calcula o valor do arco cosseno de valor entre 1 e -1. Exemplo:

```
double acos = Math.acos (1.0);  
System.out.println ( "acos =" + acos );
```

*Referência: MathAcos.java*

## Math.atan ()

O método `Math.atan()` calcula o arco tangente valor de um valor entre 1 e -1. Exemplo:

```
double atan = Math.atan (1.0);  
System.out.println ( "atan = " + atan );
```

*Referência: MathAtan.java*

## Math.sinh ()

O método `Math.sinh()` calcula o valor do seno hiperbólico de um valor entre 1 e -1. Exemplo:

```
double sinh = math.sinh (1.0);  
System.out.println ( "sinh = " + sinh );
```

*Referência: MathSinh.java*

## Math.cosh ()

O método `Math.cosh()` calcula o valor do cosseno hiperbólico de um valor entre 1 e -1.  
Exemplo:

```
double cosh = Math.cosh (1.0);  
System.out.println ( "cosh = " + cosh );
```

*Referência: MathCosh.java*

## Math.tanh ()

O método `Math.tanh()` calcula o valor da tangente hiperbólica de um valor entre 1 e -1.  
Exemplo:

```
double tanh = Math.tanh (1,0);  
System.out.println ( "tanh = " + tanh );
```

*Referência: MathTanh.java*

## Math.toDegrees ()

O método `Math.toDegrees()` converte um ângulo em radianos para graus. Exemplo:

```
double graus = Math.toDegrees (Math.PI);  
System.out.println ( "graus = " + graus );
```

*Referência: MathToDegrees.java*

## Math.toRadians ()

O método `Math.toRadians()` converte um ângulo em graus para radianos. Exemplo:

```
double radianos = Math.toRadians (180);  
System.out.println ( "radianos = " + radianos );
```

*Referência: MathToRadians.java*