



Conhecendo os poderes das classes Matemáticas

4.1 Operações matemáticas

A linguagem Java possui uma classe chamada **Math** que contém diversos métodos especializados em realizar cálculos matemáticos. Observe a seguinte sintaxe:

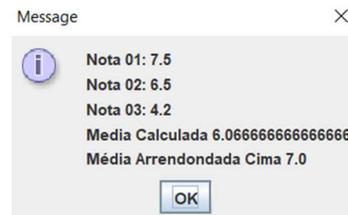
`Math.<nome do método>(<argumentos ou lista de argumentos>)`

Método ceil

O método `ceil` tem como função realizar o arredondamento de um número (do tipo `float` ou `double`) para o seu próximo inteiro, por exemplo: o próximo inteiro de 1.8 é 2, de 5.5 é 6, de 4.1 é 5, e assim por diante. Sua sintaxe é a seguinte:

`Math.ceil(<valor>)`

```
1 import javax.swing.*;
2 public class ArredondandoPraCima {
3     public static void main(String[] args) {
4         double nota1, nota2, nota3, media;
5         nota1 = Double.parseDouble(JOptionPane.showInputDialog("Digite nota 01: "));
6         nota2 = Double.parseDouble(JOptionPane.showInputDialog("Digite nota 02: "));
7         nota3 = Double.parseDouble(JOptionPane.showInputDialog("Digite nota 03: "));
8         media = (nota1 + nota2 + nota3)/3;
9         JOptionPane.showMessageDialog(null, " Nota 01: " + String.valueOf(nota1)
10            + "\n Nota 02: " + String.valueOf(nota2)
11            + "\n Nota 03: " + String.valueOf(nota3)
12            + "\n Média Calculada " + media
13            + "\n Média Arredondada Cima " + Math.ceil(media)
14        );
15    }
16 }
```

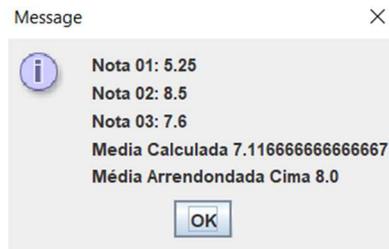


Método floor

Assim como `ceil`, o método `floor` também é utilizado para arredondar um número, mas para o seu inteiro anterior, por exemplo: o inteiro anterior de 1.1 é 1, de 2.9 é 2 e de 6.54 é 6. Sua sintaxe é a mesma do método `ceil`:

`Math.floor(<valor>)`

```
1 import javax.swing.*;
2 public class ArredondandoPraCima {
3     public static void main(String[] args) {
4         double nota1, nota2, nota3, media;
5         nota1 = Double.parseDouble(JOptionPane.showInputDialog("Digite nota 01: "));
6         nota2 = Double.parseDouble(JOptionPane.showInputDialog("Digite nota 02: "));
7         nota3 = Double.parseDouble(JOptionPane.showInputDialog("Digite nota 03: "));
8         media = (nota1 + nota2 + nota3)/3;
9         JOptionPane.showMessageDialog(null, " Nota 01: " + String.valueOf(nota1)
10            + "\n Nota 02: " + String.valueOf(nota2)
11            + "\n Nota 03: " + String.valueOf(nota3)
12            + "\n Média Calculada " + media
13            + "\n Média Arredondada Cima " + Math.ceil(media)
14        );
15    }
16 }
```



Métodos round, max, min, sqrt, pow e abs

Esta seção apresenta alguns métodos da classe Math. A Tabela 4.1 traz um resumo deles.

Tabela 4.1 – Alguns métodos da classe Math

Método	Sintaxe	Descrição
round	Math.round(<valor>)	Recebe um valor numérico e retorna esse valor arredondado. Para valores decimais <0.5 arredonda para baixo, para valores >=0.5 arredonda para cima. Exemplos: Math.round(2.35) → 2, Math.round(2.59) → 3
max	Math.max(<valor1>, <valor2>)	Recebe dois valores numéricos e retorna o maior deles. Exemplo: Math.max(10,20) → 20
min	Math.min(<valor1>, <valor2>)	Recebe dois valores numéricos e retorna o menor deles. Exemplo: Math.max(10,20) → 10
sqrt	Math.sqrt(<valor>)	Recebe um valor numérico e retorna sua raiz quadrada. Exemplo: Math.max(25) → 25
pow	Math.pow(<valor1>, <valor2>)	Recebe dois valores numéricos (o operando e o expoente) e eleva o primeiro valor ao segundo. Exemplo: Math.max(10,2) → 100
abs	Math.abs(<valor>)	Recebe um valor numérico e retorna seu valor absoluto, desconsiderando o sinal. Exemplo: Math.max(-2) → 2

```
1 import javax.swing.*;
2 public class Varios_Metodos {
3     public static void main(String[] args) {
4         double a = 2.4 , b = 2.8 , c = 4, d = -1 ;
5         JOptionPane.showMessageDialog(null, "Valor de a "+a
6         + "\n Valor de b " + b
7         + "\n Valor de c " + c
8         + "\n Valor de d " + d
9         + "\nArrendodamento de (a) " + Math.round(a)
10        + "\nArrendodamento de (b) " + Math.round(b)
11        + "\nMaior valor entre (a) e (b) " + Math.max(a,b)
12        + "\nMenor valor entre (c) e (d) " + Math.min(c,d)
13        + "\nRaiz quadrada de (c) " + Math.sqrt(c)
14        + "\n(c) elevado ao cubo c³ " + Math.pow(c,3)
15        + "\nValor absoluto de (d) " + Math.abs(d)
16        );
17    }
18 }
```

Message



Valor de a 2.4
Valor de b 2.8
Valor de c 4.0
Valor de d -1.0
Arrendodamento de (a) 2
Arrendodamento de (b) 3
Maior valor entre (a) e (b) 2.8
Menor valor entre (c) e (d) -1.0
Raiz quadrada de (c) 2.0
(c) elevado ao cubo c³ 64.0
Valor absoluto de (d) 1.0

OK

USANDO DECIMAL FORMAT

```
1 import java.text.DecimalFormat;
2
3 public class Usando_Decimal {
4     public static void main(String[] args) {
5         double num = 7.63/4.85;
6         System.out.println(num);
7         DecimalFormat dec = new DecimalFormat();
8         dec.applyPattern("0.00");
9         double valor= 1235133580.25;
10        System.out.println(dec.format(num));
11        dec.applyPattern("R$ 0,000.00");
12        System.out.println(dec.format(valor));
13    }
14 }
```

```
1.57319587628866
```

```
1,57
```

```
R$ 1.235.133.580,25
```

```
Process finished with exit code 0
```