

Java

O Método “**abs**” retorna o valor **Absoluto**.

abs	Valor Absoluto	<code>Math.abs(-10)</code>	10
------------	----------------	----------------------------	----

É possível forçar o **arredondamento para baixo** ou **Trucagem** ou **Divisão Inteira**.

Neste caso, ao arredondarmos o valor 3,9 o resultado será 3 e NÃO 4; mas CUIDADO, ao arredondar o valor -7,3 o resultado será -8, pois este é o valor inteiro abaixo mais próximo.

abs	Valor Absoluto	<code>Math.abs(-10)</code>	10
floor	Arredonda para Baixo	<code>Math.floor(3.9)</code>	3

Já o Método “**ceil**” faz exatamente o contrário do Método “floor”, ou seja, arredonda o valor para cima.

abs	Valor Absoluto	<code>Math.abs(-10)</code>	10
floor	Arredonda para Baixo	<code>Math.floor(3.9)</code>	3
ceil	Arredonda para Cima	<code>Math.ceil(4.2)</code>	5

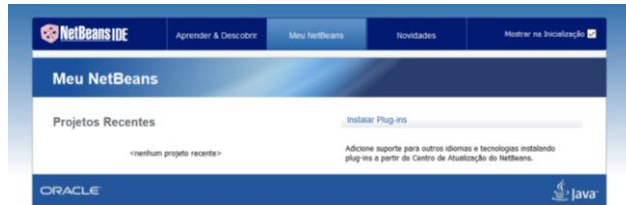
Já o Método “**round**” é o método tradicional, ou seja, o método aritmético que utilizamos para arredondar pela aproximação. Neste caso 5,6 é mais próximo de 6.

abs	Valor Absoluto	<code>Math.abs(-10)</code>	10
floor	Arredonda para Baixo	<code>Math.floor(3.9)</code>	3
ceil	Arredonda para Cima	<code>Math.ceil(4.2)</code>	5
round	Arredonda Aritmeticamente	<code>Math.round(5.6)</code>	6

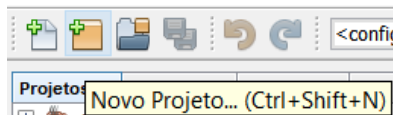
Exercício

Siga os passos abaixo para realizar tarefas sobre arredondamentos.

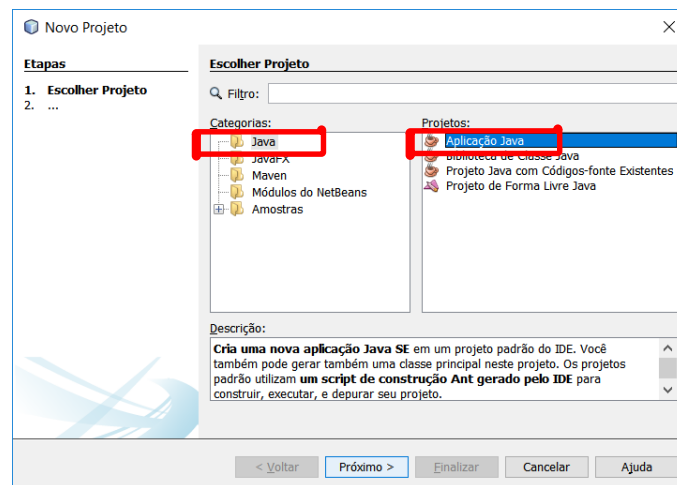
Abra o NetBeans.



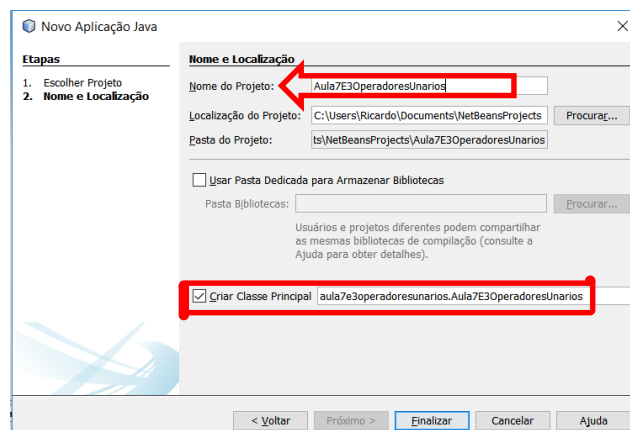
Crie um Novo Projeto.



Em Categoria mantenha Java selecionado e em Projeto, mantenha Aplicação Java e clique no botão Próximo.

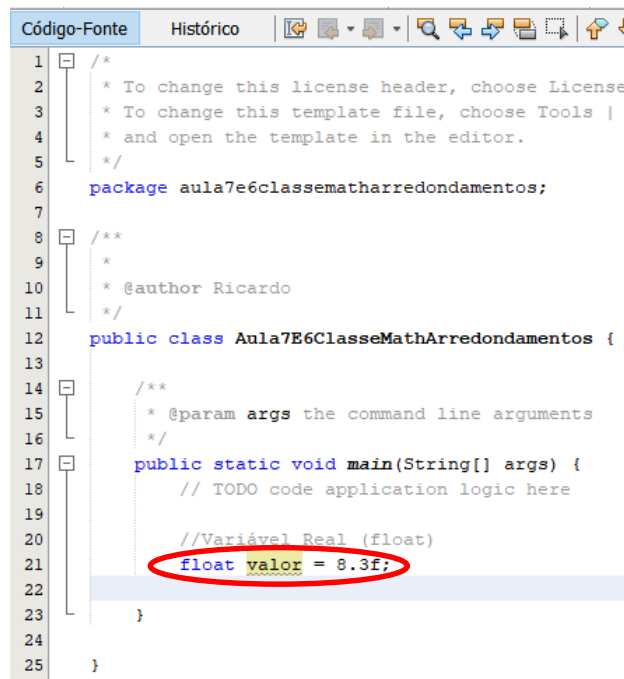


Nomeie o Projeto, mantenha “Criar Classe Principal” selecionada e clique em Finalizar.



Crie uma **Variável Real (float)** com valor 8,3.

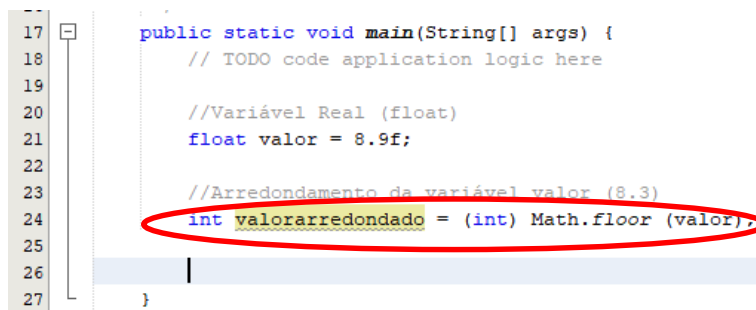
Float valor = 8.3f;



```
1  /*
2  * To change this license header, choose License
3  * To change this template file, choose Tools |
4  * and open the template in the editor.
5  */
6  package aula7e6classematharredondamentos;
7
8  /**
9   *
10  * @author Ricardo
11  */
12  public class Aula7E6ClasseMathArredondamentos {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19
20          //Variável Real (float)
21          float valor = 8.3f;
22
23      }
24
25  }
```

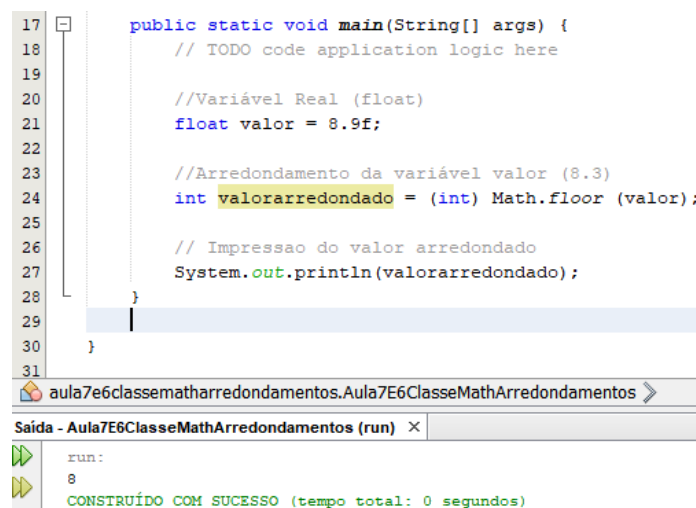
Faça o **arredondamento deste valor para baixo**, ou seja, force utilizando o Método “**floor**”.

int valorarredondado = (int) Math.floor (valor);



```
17  public static void main(String[] args) {
18      // TODO code application logic here
19
20      //Variável Real (float)
21      float valor = 8.9f;
22
23      //Arredondamento da variável valor (8.3)
24      int valorarredondado = (int) Math.floor (valor);
25
26
27  }
```

Imprima o valor arredondado utilizando: **System.out.println(valorarredondado);**



```
17  public static void main(String[] args) {
18      // TODO code application logic here
19
20      //Variável Real (float)
21      float valor = 8.9f;
22
23      //Arredondamento da variável valor (8.3)
24      int valorarredondado = (int) Math.floor (valor);
25
26      // Impressao do valor arredondado
27      System.out.println(valorarredondado);
28  }
29
30  }
```

aula7e6classematharredondamentos.Aula7E6ClasseMathArredondamentos

Saída - Aula7E6ClasseMathArredondamentos (run) X

run:
8
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

Para **arredondar esse valor para cima**, substitua o Método “floor” pelo **Método “ceil”**.

Int valorarredondado = (int) Math.ceil (valor);

```
17 public static void main(String[] args) {
18     // TODO code application logic here
19
20     //Variável Real (float)
21     float valor = 8.9f;
22
23     /*
24     1 Exercício: MÉTODO FLOOR
25
26     //Arredondamento da variável valor (8.3) para BAIXO
27     int valorarredondado = (int) Math.floor (valor);
28     */
29
30     //Arredondamento da variável valor (8.3) para CIMA
31     int valorarredondado = (int) Math.ceil (valor);
32
33     // Impressao do valor arredondado
34     System.out.println(valorarredondado);
35 }
```

aula7e6classematharredondamentos.Aula7E6ClasseMathArredondamentos > main

Saída - Aula7E6ClasseMathArredondamentos (run) X

```
run:
9
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

No caso da imagem utilizada como exemplo, a resposta do arredondamento foi similar ao **ARREDONDANDO PELO MÉTODO ARITMÉTICO**, ou seja, por **Aproximação**. Para verificar, altere o **Método** para “**round**”.

Int valorarredondado = (int) Math.round (valor);

```
38 //Arredondamento ARITMÉTICO da variável valor (8.3
39 int valorarredondado = (int) Math.round (valor);
40
41 // Impressao do valor arredondado
42 System.out.println(valorarredondado);
```

aula7e6classematharredondamentos.Aula7E6ClasseMathArredondamentos > main

Saída - Aula7E6ClasseMathArredondamentos (run) X

```
run:
9
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

Altere o valor de 8,9 para 8,5 e verifique que o resultado continuará sendo 9, mas alterando o valor para 8,4 o resultado passará a ser 8.

```
17 public static void main(String[] args) {
18     // TODO code application logic here
19
20     //Variável Real (float)
21     float valor = 8.5f;
22
23     /*
24     1 Exercício: MÉTODO FLOOR
25
26     //Arredondamento da variável valor (8.3) para BAIXO
27     int valorarredondado = (int) Math.floor (valor);
28     */
29
30     /*
31     2 Exercício: MÉTODO CEIL
32
33     //Arredondamento da variável valor (8.3) para CIMA
34     int valorarredondado = (int) Math.ceil (valor);
35     */
36
37     //Arredondamento ARITMÉTICO da variável valor (8.3)
38     int valorarredondado = (int) Math.round (valor);
39
40     // Impressao do valor arredondado
41     System.out.println(valorarredondado);
42 }
```

aula7e6classematharredondamentos.Aula7E6ClasseMathArredondamentos > main

Saída - Aula7E6ClasseMathArredondamentos (run) X

```
run:
9
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

```
17 public static void main(String[] args) {
18     // TODO code application logic here
19
20     //Variável Real (float)
21     float valor = 8.4f;
22
23     /*
24     1 Exercício: MÉTODO FLOOR
25
26     //Arredondamento da variável valor (8.3) para BAIXO
27     int valorarredondado = (int) Math.floor (valor);
28     */
29
30     /*
31     2 Exercício: MÉTODO CEIL
32
33     //Arredondamento da variável valor (8.3) para CIMA
34     int valorarredondado = (int) Math.ceil (valor);
35     */
36
37     //Arredondamento ARITMÉTICO da variável valor (8.3)
38     int valorarredondado = (int) Math.round (valor);
39
40     // Impressao do valor arredondado
41     System.out.println(valorarredondado);
42 }
```

aula7e6classematharredondamentos.Aula7E6ClasseMathArredondamentos > main

Saída - Aula7E6ClasseMathArredondamentos (run) X

```
run:
8
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```