

Mão na massa: algumas conversões em Java

Vamos agora praticar algumas conversões em Java, para isso, siga os passos abaixo:

1) Crie uma classe e escreva seu método main, no curso escolhemos o nome "**TestaConversao**".

2) Escreva o seguinte código:

```
public class TestaConversao{  
  
    public static void main(String[] args){  
  
        double salario = 1270.50;  
        int valor = salario;  
    }  
}
```

Esse código não compila, uma vez que estamos tentando atribuir uma variável **double** dentro de uma variável **int**, o que não é possível na linguagem Java.

Lembre-se, entretanto, que o inverso é válido, ou seja, podemos atribuir um valor do tipo **int** dentro de uma variável do tipo **double**.

3) Existe uma forma de fazer com que esse código compile. Para isso usaremos o que chamamos de **casting**, para isso, modifique seu código e adicione **(int)** antes da variável **salario**:

```
public class TestaConversao{  
  
    public static void main(String[] args){  
  
        double salario = 1270.50;  
        int valor = (int) salario;  
    }  
}
```

Quando fazemos o **casting**, estamos pedindo ao Java que ache uma forma de transformar o valor no lado direito do **(int)**, no nosso caso, a variável **salario**, em inteiro. Dessa forma o java transformará **double** para **int**.

4) "Printe" a variável **valor** no console, para isso, adicione:

```
System.out.println(valor);
```

A saída será **1270**.

Basicamente, para as variáveis primitivas, o **casting** nada mais é do que fazer com que essa conversão seja feita quando ela não é feita de maneira automática.

5) No momento, nossa variável **salario** é do tipo **double**, será que essa é a melhor escolha? Teste o seguinte:

Escreva 3 variáveis do tipo **double** como feito abaixo, a última variável (**total**) receberá o somatório das outras duas, observe:

```
public class TestaConversao{

    public static void main(String[] args){

        double salario = 1270.50;
        int valor = (int) salario;
        System.out.println(valor);

        double valor1 = 0.2;
        double valor2 = 0.1;
        double total = valor1 + valor2;

        System.out.println(total);
    }
}
```

Imaginamos que nossa saída seria **0.3**, mas obtemos um número completamente diferente.

Devido essa inconsistência, **double** não seria a melhor escolha para guardarmos salário. Existem outras alternativas melhores que veremos em outros cursos.

6) Vamos testar também o tipo **float**, para isso escreva a variável **pontoFlutuante** do tipo **float**, veja abaixo:

```
public class TestaConversao{

    public static void main(String[] args){

        float pontoFlutuante = 3.14; // aqui!

        double salario = 1270.50;
        int valor = (int) salario;
        System.out.println(valor);

        double valor1 = 0.2;
        double valor2 = 0.1;
        double total = valor1 + valor2;

        System.out.println(total);
    }
}
```

Dessa forma, teremos um erro, já que para o Java, **3.14** é um **double**. Esse é o mesmo problema que tivemos quando tentamos guardar um **double** dentro de um **int**. No tipo **double** podemos armazenar 64 bits, no **float** e no **int** apenas 32 bits. Isso pode acarretar em perda de informação, para que esse código funcione, podemos fazer o **casting**, colocando **(float)** na frente de **3.14**, mas faremos de uma outra forma.

7) Informe ao Java que o valor **3.14** é um **float**, para isso coloque **f** ao final do número, nosso código ficará assim:

```
public class TestaConversao{  
  
    public static void main(String[] args){  
  
        float pontoFlutuante = 3.14f;  
  
        double salario = 1270.50;  
        int valor = (int) salario;  
        System.out.println(valor);  
  
        double valor1 = 0.2;  
        double valor2 = 0.1;  
        double total = valor1 + valor2;  
  
        System.out.println(total);  
    }  
}
```

Agora nosso código compila!