



# LISTA DE EXERCÍCIOS 8 – TIPOS DE DADOS E CONVERSÕES

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HERCULANO DE BIASI

[herculano.debiasi@unoesc.edu.br](mailto:herculano.debiasi@unoesc.edu.br)



# LISTA DE EXERCÍCIOS 8

1. Crie um programa que declare uma variável do tipo `int` com o valor 12 e utilize conversão explícita (*casting*) para convertê-la para um tipo `byte`.  
A seguir crie uma variável `double` com o valor 1234.56 e utilize conversão explícita para convertê-la para um tipo `float`. Imprima na tela o resultado das conversões, ou seja, as variáveis `byte` e `float`.
2. Corrija as linhas 5 e 8 do código abaixo utilizando o conceito de *casting* explícito. Acrescente o código para imprimir as variáveis `numero2` e `numero4`.

```
1 public class Teste {  
2  
3     public static void main(String[] args) {  
4         double numero1 = 1234.56;  
5         float numero2 = numero1;  
6  
7         int numero3 = 12;  
8         byte numero4 = numero3;  
9     }  
10  
11 }
```

# LISTA DE EXERCÍCIOS 8

3. Em todas as questões abaixo, use a classe `Math` do Java para calcular e imprimir

- a)  $2^{10}$ , dois elevado à décima potência
- b)  $\sqrt{1000}$ , raiz quadrada de 1000
- c) Valor absoluto (sem sinal) de -10
- d) Calcular o valor máximo entre os números 42 e 666
- e) Calcular o valor mínimo entre os números 42 e 666
- f) Arredondar o número 4,9 para cima (método `ceil`)
- g) Arredondar o número 4,9 para baixo (método `floor`)
- h) Calcular o seno (`sin`) e cosseno (`cos`) de  $90^\circ$  e a tangente (`tan`) de  $45^\circ$ 
  - i. Esse é um processo mais complexo; o seno de 90 graus é 1, o cosseno de 90 graus é 0 e a tangente de 45 graus também é 1
  - ii. O primeiro passo é converter o número de graus para radianos com o método `toRadians()`, pois as funções trigonométricas trabalham com radianos e não com graus
  - iii. O segundo passo é usar o valor em radianos para calcular o seno, cosseno e tangente
  - iv. Finalmente, arredondar o resultado com o método `round()`

# LISTA DE EXERCÍCIOS 8

4. Ainda usando a classe `Math` do Java faça um programa que calcule o IMC (Índice de Massa Corporal) segundo a fórmula abaixo:

$$IMC = \frac{Peso}{Altura^2}$$

- a) O peso e a altura deverão ser solicitados ao usuário (usem `Scanner`)
- b) O programa deverá imprimir o índice calculado e a classificação de acordo com a seguinte tabela (usem `if()`)

IMC	CLASSIFICAÇÃO
MENOR QUE 18,5	MACREZA
ENTRE 18,5 E 24,9	NORMAL
ENTRE 25,0 E 29,9	SOBREPESO
ENTRE 30,0 E 39,9	OBESIDADE
MAIOR QUE 40,0	OBESIDADE GRAVE

# LISTA DE EXERCÍCIOS 8

5. Sem executar o código tente determinar qual é o valor que será impresso. Após isso execute o código para verificar se sua resposta estava correta.

```
modulo2 - Exercicios.java

1 public class Exercicios {
2     public static void main(String[] args) {
3         int raio = 5000;
4         int circunferencia = (int) (2 * Math.PI * raio);
5         System.out.println(circunferencia);
6     }
7 }
8
```

- a) 31416
- b) 30000
- c) 31415
- d) 314159
- e) 31415.9

# LISTA DE EXERCÍCIOS 8

6. O código abaixo, estranhamente, está mostrando o resultado 1 (valor inteiro) em vez de 1.25 (valor em ponto-flutuante). Modifique a linha 4 de forma que seja feita uma conversão implícita para `float` e na linha 5 faça um `cast` explícito.

```
modulo2 - Exercicios.java

1 public class Teste {
2
3     public static void main(String[] args) {
4         System.out.println("10/8 = " + 10/8);
5         System.out.println("10/8 = " + (...) 10/8);
6     }
7
8 }
```

# LISTA DE EXERCÍCIOS 8

7. Em Java, constantes são definidas com a palavra-chave `final`. A palavra-chave `static`, por sua vez, indica que o atributo é acessado através da classe em si e não através de suas instâncias (objetos). Complete o código de forma a imprimir o conteúdo das constantes abaixo.

```
modulo2 - TesteConstantes.java

1  public class TesteConstantes {
2
3      final static double PI = 3.1415;
4      final long NUMERO = 999_999L;
5
6      public static void main(String[] args) {
7          final String NOME = "DEV-TI";
8
9          System.out.println(); // Imprima aqui a constante PI
10         System.out.println(); // Imprima aqui a constante NUMERO
11         System.out.println(); // Imprima aqui a constante NOME
12     }
13
14 }
```

# LISTA DE EXERCÍCIOS 8

8. Use o recurso de *autoboxing* do Java para atribuir as variáveis primitivas `num1`, `num2` e `num3` às suas respectivas classes *wrapper*.

```
modulo2 - TesteWrappers.java
1  public class TesteWrappers {
2
3      public static void main(String[] args) {
4          int num1 = 11;
5          float num2 = 22.22f;
6          double num3 = 33.33;
7
8          // Aplicação de wrappers e auto boxing
9          Integer w1 = ...
10         Float w2 = ...
11         Double w3 = ...
12
13         System.out.println(w1);
14         System.out.println(w2);
15         System.out.println(w3);
16     }
17
18 }
```



# LISTA DE EXERCÍCIOS 8

9. Use o método `valueOf()` da classe `String` nas linhas 5 e 9 para transformar os dados numéricos em *strings*.

```
modulo2 - TesteConversoes.java

1  public class TesteConversoes {
2
3      public static void main(String[] args) {
4          Integer num1 = 123;
5          String valor1 = ... ;
6          System.out.println(valor1);
7
8          double num2 = 789.01;
9          String valor2 = ... ;
10         System.out.println(valor2);
11     }
12
13 }
```

# LISTA DE EXERCÍCIOS 8

10. Use os métodos `parseInt()` e `parseDouble()` das classe `Integer` e `Double` nas linhas 10 e 19 para transformar as *string* em formatos numéricos. Complete o restante do código como solicitado nos comentários abaixo.

```
modulo2 - TesteConversoes.java

1 import java.text.NumberFormat;
2 import java.util.Locale;
3
4 public class TesteConversoes {
5     public static void main(String[] args) {
6         // Defina abaixo o Locale como US
7         Locale.setDefault(...);
8
9         String str1 = "123456";
10        int num1 = ... ;
11
12        // Imprima num1 usando NumberFormat incluindo o símbolo monetário
13        System.out.println(...);
14
15        // Defina abaixo o Locale como português do Brasil
16        Locale.setDefault(...);
17
18        String str2 = "789.01";
19        double num2 = ... ;
20
21        // Imprima num2 usando NumberFormat incluindo o símbolo monetário
22        System.out.println(...);
23    }
24 }
```

# LISTA DE EXERCÍCIOS 8

- II. Adicione, no exemplo feito do sistema de hotel, no mínimo mais três *interfaces* e três classes que implementem essas *interfaces*; se necessário ou adequado, utilize essas *interfaces* recém-criadas também nas classes já existentes. Teste essas classes no programa principal.

Exemplos de *interfaces*:

- ICozinha
- IFazTudo
- IAuxiliarGeral