

## Aula 2 - Variáveis e um pouco mais que imprimir linhas na tela

Na aula anterior, vimos nossa primeira classe Java: uma classe que apenas exibe uma mensagem na tela:

```
package pacote;

public class PrimeiraClasse {

    public static void main(String[] args) {
        System.out.println("Meu primeiro programa em
Java!!");
    }
}
```

Evidentemente que um programa desses não tem muitas utilidades. Então, vamos ver mais alguns recursos que uma classe Java pode ter.

### Variáveis primitivas

As variáveis primitivas nos ajudam a armazenar valores, que serão necessários para realizarmos todo o tipo de computação.

Em Java, devemos sempre declarar o tipo da variável antes do nome dela. Algo como:

```
tipoDaVariavel nomeDaVariavel;
```

Imaginem que precisamos armazenar os dados de um funcionário. Um funcionário tem um nome, uma idade, um salário e o seu sexo.

Podemos pensar em variáveis como:

```
tipoDaVariavel idade;
tipoDaVariavel salario;
tipoDaVariavel sexo;
tipoDaVariavel casado;
tipoDaVariavel nome;
```

Mas em Java, cada uma dessas variáveis tem um tipo diferente. Então, vamos colocar os nomes certos nas variáveis, começando pela `idade`:

```
int idade;
```

Uma variável `int` é uma variável do tipo inteira. Ou seja, só aceita valores inteiros.

Podemos inclusive atribuir valores às variáveis:

```
int idade = 43;
```

Podemos ler a linha de cima como "a variável `idade` é do tipo `int` e possui o valor 43".

Agora vamos ver a variável `salário`.

Normalmente o salário de alguém é um valor que possui centavos também. Então, armazenarmos o salário numa variável do tipo `int` não é uma boa ideia, pois a pessoa que irá recebê-lo deixará de receber os centavos do seu salário.

Em Java, existe um tipo primitivo que permite o armazenamento de valores reais (ou seja, não inteiros): o `double`.

No nosso caso, podemos escrever:

```
double salario;
```

E se quisermos atribuir um valor a ele:

```
double salario = 4532.51;
```

Para o sexo da pessoa, podemos trabalhar com letras:

- M para masculino
- F para feminino
- N para não declarado

Nesse caso, podemos usar o tipo `char`. E a variável então ficará assim (já valorada):

```
char sexo = 'M';
```

Notem as aspas simples (') antes e depois da letra.

*Diversos caracteres que escrevemos não somente em Java, mas em qualquer linguagem de programação (como { (abre chaves), } (fecham chaves), " (aspas duplas) ou mesmo ' (aspas simples, citada acima)) demandam o uso da tecla shift para que ela seja escrita. Logo, é importante saber sua localização (do lado esquerdo e direito do teclado).*

Temos agora mais 2 variáveis: `casado` e (finalmente) o `nome`.

Existem apenas 2 possibilidades para a variável `casado`: `sim` ou `não`.

Em Java, podemos usar o tipo `boolean` para esse tipo de informação. E somente 2 valores podem ser inseridos em variáveis com essas:

```
boolean casado = true;
```

ou

```
boolean casado = false;
```

onde `true` representa o "sim", ou algo verdadeiro, e `false` representa o "não", ou falso.

Para o nome, usamos o tipo `String`.

*Uma String é um pouco mais que uma variável primitiva, mas falaremos mais sobre isso ao longo do curso. Por hora, vamos considerar como tal.*

Então, devemos realizar a alteração:

```
String nome = "Antonio das dores";
```

Resumindo, os principais tipos primitivos do Java são essas:

Tipo Primitivo	Quando usar?
<code>int</code>	Para manipulação de valores inteiros
<code>double</code>	Para trabalhar com valores reais
<code>char</code>	Somente letras. Para textos, usar <code>String</code>
<code>boolean</code>	Informação do tipo "sim" ou "não", "verdadeiro" ou "falso"
<code>String</code>	Armazenamento de textos

## Operações com Tipos Primitivos

Podemos realizar operações com os tipos. Por exemplo, os tipos "numéricos" podem ser subtraídos, somados, multiplicados ou divididos.

```
int umaVariavel = 45 * 3;
```

```
int outraVariavel = 65;
int maisOutraVariavel = umaVariavel + outraVariavel;
```

Podemos fazer o mesmo com valores `double`:

```
double salario = 4500;
double porcentagemDeAumento = 10;
double salarioReajustado = salario + (salario / 100 *
porcentagemDeAumento);
```

Para rodar esses códigos, basta que eles estejam dentro do bloco `main`:

```
public class VariaveisPrimitivas {

    public static void main(String[] args) {
        int idade = 43;
        double salario = 4532.50;
        char sexo = 'M';
        boolean casado = true;
        String nome = "Antonio das Dores";

        int umaVariavel = 45 * 3;
        int outraVariavel = 65;
        int maisOutraVariavel = umaVariavel +
        outraVariavel;

        double porcentagemDeAumento = 10;
        double salarioReajustado = salario + (salario / 100
        * porcentagemDeAumento);

        System.out.println(idade);
        System.out.println(salario);
        System.out.println(sexo);
        System.out.println(casado);
        System.out.println(nome);

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

Notaram o `System.out.println()` ? O "ln" no final do nome do comando é uma abreviação de *line*, e faz com que cada informação seja colocada numa linha diferente.

## Exercício (OBS: crie um projeto somente para ele no Eclipse)

1 - Uma transportadora precisa comprar veículos utilitários de transporte, entre eles um caminhão simples, um bitrem, um furgão e um VUC (veículo urbano de carga). Sabendo que eles custam respectivamente R\$200.000,00, R\$400.000,00, R\$100.000,00 e R\$100.000,00, escreva um programa que calcule e imprima o total a ser gasto pela empresa.

Dica: crie variáveis para cada um dos veículos, crie outra variável onde seja feita a soma e, ao final, imprima a variável onde foi feita a soma.