

Assunto: Definição de Classes, atributos e métodos, criação (instaciação) de Objetos, entrada/saída de dados na Linguagem Java

Orientação a objeto é uma maneira de programar que nos ajuda na organização e soluciona problemas enfrentados no mundo procedural.

Classes

Uma classe no JAVA representa nada mais que um modelo ou um gabarito de algo do mundo real que se queira reproduzir no ambiente de desenvolvimento.

Pensando desta forma é muito fácil entender e pensar sobre como se projetar um sistema com orientação a objetos.

Quando pensamos em desenvolver um sistema voltado a locação de veículos, por exemplo, a primeira classe que vem à tona é uma que possa representar a figura do carro no seu sistema, daí temos então nossa primeira classe. Uma classe é composta basicamente de 3 itens:

- Nome da Classe - responsável por identificar a classe deve ser utilizado nas referências aos objetos da classe.

Por convenção (boas práticas) a classe deve ter a 1ª letra em maiúscula no nome da classe e as demais minúsculas. Caso o nome da classe seja composto por mais de uma palavra, coloque sempre a primeira letra de cada palavra em letra maiúscula.

Lembre-se também de nomear o arquivo com o mesmo nome da classe e respeite a convenção pois a linguagem Java é *case-sensitive* (tem diferenciação entre maiúsculo de minúsculo)

<pre>class Veiculo { } ou class VeiculoAnfibio{ }</pre>	<div style="display: inline-block; vertical-align: middle; text-align: center;"><div style="border-top: 1px solid black; width: 100px; margin: 0 auto;"></div><div style="font-size: 2em; margin: 0 auto;">→</div></div> <div style="display: inline-block; vertical-align: middle;"><p>Nome arquivo – Veiculo.java</p><p>Nome arquivo – VeiculoAnfibio.java</p></div>
---	--

- Atributos - São valores que possam representar as propriedades e/ou características possíveis que os objetos desta classe podem assumir. Por convenção, costuma-se escrever o atributo com letras minúsculas, a menos que ele seja composto por mais de uma palavra. Neste caso, a mesma regra citada nos nomes de Classe também é válida.

```
class Veiculo {  
    int anoModelo;  
    int anoFabricacao  
    String cor;  
}
```

- Métodos - representam as funcionalidades que a classe apresentará. Podemos dizer que a classe tem um ou mais comportamento. Utilize os nomes dos métodos com letras

minúsculas, a menos que ele seja composto por mais de uma palavra. Neste caso, a mesma regra citada nos nomes de Classe também é válida.

```
class Veiculo {  
    int anoModelo;  
    int anoFabricacao  
    String cor;  
    void alteraCor (String cor){  
        this.cor = cor;  
    }  
}
```

Não se preocupe com itens não explicados pois eles fazem parte da disciplina e serão posteriormente explicados.

Objetos

Um objeto é a representação de uma classe. Vários objetos podem ser criados utilizando-se a mesma classe. Basta pensarmos na Classe como um tipo de dado ou uma forma onde todos os objetos serão construídos conforme a forma ou conforme o tipo de dado definido, possuindo todas as características da classe.

Na linguagem JAVA, para criarmos um novo objeto, basta utilizar o comando **new** da seguinte forma:

```
Veiculo meuVeiculo = new Veiculo();
```

Seguindo as boas práticas utilizaremos o seguinte padrão para a construção de uma classe:

```
public class NovaClasse{  
  
    public int atributo1;  
    public int atributo2;  
    public boolean atributo3;  
  
    public void metodo1(){  
        comandos  
    }  
    public void metodo2(){  
        comandos  
    }  
}
```

} **Declaração dos atributos**

} **Declaração dos métodos**

É importante ressaltar que na linguagem JAVA essa ordem de declarações não é obrigatória, porém mais uma vez, é conveniente para melhorar a legibilidade do código, padronização e proximidade ao diagrama de classes da UML.

Métodos

Em Java os métodos podem retornar ou não valores.

Com retorno

Para que seja possível realizar o retorno do método, primeiramente é necessário declarar qual o tipo de retorno será realizado pelo método, para isso devemos utilizar a seguinte sintaxe:

```
public int meuMetodo(){  
}
```

Na declaração acima significa que pretendemos que este método retorne para quem o chamou um valor do tipo `int`. Porém, para que o método realmente retorne algum valor, é necessário agora pontuarmos qual valor será retornado. Para tal, faremos uso da palavra reservada **return**, da seguinte forma:

```
public int meuMetodo(){  
return 1;  
}
```

Com isso temos agora um método que retornará o valor 1 do tipo `int`.

É importante lembrar que, uma vez declarado um retorno para o método, é obrigatório que este retorno aconteça. O fato de não se enviar um retorno reflete em um erro de execução.

Métodos com recebimento de parâmetro

Na linguagem JAVA métodos também são capazes de receber um ou mais valores que podem ou não ser utilizados em seu processamento. Para tal, devemos declarar quais os tipos de parâmetros serão recebidos. Como por exemplo:

```
public void meuMetodo (int numero){  
  
comandos;  
  
}
```

ou ainda com mais de um parâmetro....

```
public void meuMetodo2 (int numero, boolean verdade){  
  
    comandos;  
  
}
```

Quando fazemos uma chamada a um método com parâmetros de entrada, um erro na passagem dos tipos dos parâmetros representa um erro de compilação.

Exercício

Não se esqueça de criar uma classe para o televisor e uma outra classe que manipulará a classe televisor. Resumindo, você deverá ter 2 arquivos com a extensão .java. Uma para a classe televisor e outro para a utilizar o televisor. Não se esqueça da reusabilidade. Crie um roteiro para testar a 2ª classe, que utilizará o televisor, em Word.

Crie uma classe Televisor. Essa classe deve possuir três atributos:

- canal (inicia em 1 e vai até 16)
- volume (inicia em 0 e vai até 10)
- ligado (inicia em desligado ou false)

com os seguintes métodos:

- aumentarVolume()
- reduzirVolume()
- subirCanal()
- descerCanal()
- ligarTelevisor()
- desligarTelevisor()
- mostraStatus() // dizer qual o canal, o volume e se o televisor está ligado

Feito isto, crie uma classe principal que utilize um novo objeto do tipo Televisor e que deve realizar as seguintes operações:

- Ligue o televisor;
- Suba 5 canais;
- Aumente o volume até o máximo;
- Mostre o Status;
- Desça até o primeiro canal;
- Reduza 2 pontos no volume;
- Mostre o Status;