Programação Orientada a Objetos

Objetos

Professor Me.: Gustavo Siqueira Vinhal

Objetos

Objetos: Uma instância de uma classe.

- Vários objetos que são instâncias de uma mesma classe têm acesso aos mesmos métodos;

- Cada objeto possui um conjunto de atributos e métodos.

- Atributos
  - Raça: Poodle
  - Nome: Rex
  - Peso: 5 quilos
- Método
  - Latir
  - Comer
  - Dormir

- Atributos
  - Potência: 500cc
  - Modelo: Honda
  - Ano: 1998
- Método
  - Acelerar
  - Frear
  - Abastecer

Criando objetos

- A criação (instanciação) de um novo objeto se dá pelo comando **new**

```
Nome da classe  Nome do objeto = new Nome da classe( ) ;
```

Exemplos:

```
VolcanoRobot robbie = new VolcanoRobot();
URL address = new URL("http://www.go.senac.br");
```

Criando objetos

- O número e tipo dos argumentos que podem ser usados dentro dos parênteses com **new** são definidos dentro da própria classe, em um método especial chamado **construtor**.

- **Construtor:** método especial para criação e inicialização de uma nova instância de uma classe.

- Não há necessidade de destruir os objetos!

Acessando e definindo valores

- Os atributos do objeto são representados por variáveis;
- Essas variáveis podem ser de instância e de classe:
  - **Variáveis de instância**: atributos exclusivos de cada objetos. A alteração dos valores afeta apenas ao objeto em questão;
  - **Variáveis de classe**: atributos comuns aos objetos de uma mesma classe. A alteração dos valores afeta todos os objetos daquela classe.
- O objeto pode acessar seus atributos através da notação de um **ponto**;

Acessando e definindo valores

- Para definir uma variável de instância:

```
tipo nome da variável;
```

- Para definir uma variável de classe deve-se usar a diretiva **static**.

```
static tipo nome da variável;
```

Acessando e definindo valores

- Para acessar as variáveis utiliza-se:

```
Nome do objeto.atributo;
```

Exemplos:

```
float total = consumidor.valorTotal;
int idade = aluno.idade;
```

Acessando e definindo valores

- O valor retornado de uma variável pode ser um outro objeto:

  Exemplo:

  boolean check = consumidor.valorTotal.pago;

Acessando e definindo valores

- Para alterar as variáveis utiliza-se:

Nome do objeto.atributo = novo valor;

Exemplos:

consumidor.valorTotal = 3.14;
aluno.idade = 25;

Acessando e definindo valores

- Quanto utiliza-se uma variável de classe, pode-se utilizar o nome do objeto ou o nome da classe:

```
Nome do objeto.atributo;
Nome da classe.atributo;
```

```
Nome do objeto.atributo = novo valor;
Nome da classe.atributo = novo valor;
```

- Recomenda-se utilizar o nome da classe, para melhor organização do código.

Chamando métodos

- Chamar um método é semelhante a referir à uma variável do objeto: a notação utilizada é um ponto.

```
Nome do objeto.método( );
```

Chamando métodos

- Assim como as variáveis de classe, existem os métodos de classe.
- Esses tipos de métodos aplicam-se à classe como um todo e não às suas instâncias.
  - Ou seja, você não precisa instanciar um objeto de uma classe para usar esse tipo de método.

```
Nome da Classe.método( );
```

Exemplos:

```
int maiorPreco = Math.max(primeiroValor, SegundoValor);
double raizQuadrada = Math.sqrt(25);
```

Casting e conversão de objetos

- Casting é o processo de produzir um novo valor que possui um tipo diferente de sua origem.

- Existem três formas:

  - Casting entre tipos primitivos;
  - Casting entre instâncias de uma classe;
  - Casting entre tipos primitivos para objetos.

Casting e conversão de objetos

- Casting entre tipos primitivos:

```
(NomeDoTipo)valor;
```

Exemplo:

```
int valor = (int)(x/y);
```

Comparando valores e classes de objetos

- Existem três tarefas comuns entre objetos:
  - Comparar objetos;
  - Descobrir a classe de qualquer objeto;

Comparando valores e classes de objetos

- Existem três tarefas comuns entre objetos:
  - Comparar objetos:
- Dos operadores de comparação, somente o igualdade (==) e diferença (!=) podem ser utilizados para comparação entre objetos. MAS....
  - Essas comparações apenas verificam se os objetos são os mesmos.
  - Para comparação de valores, deve-se criar métodos que realizem essas atividades.

Comparando valores e classes de objetos

- Existem três tarefas comuns entre objetos:
  - Descobrir a classe de qualquer objeto:

```
Objeto.getClass( ).getName( );
```

- getClass() é um método da classe Object;
- O retorno do método getName() é uma string contendo o nome da classe do objeto.

Prática:

1 – Crie uma classe do tipo formas geométricas com as características:

- Variáveis de instância:
  - largura (do tipo float);
  - altura (do tipo float);
- Métodos:
  - Calcular a área;
  - Calcular perímetro;
  - Mostrar atributos (todo os atributos, em inteiros).

2 – Crie uma classe que utilize essa outra classe (classe main).

3 – Estancie objetos da classe formas geométricas e acesse os atributos e métodos.

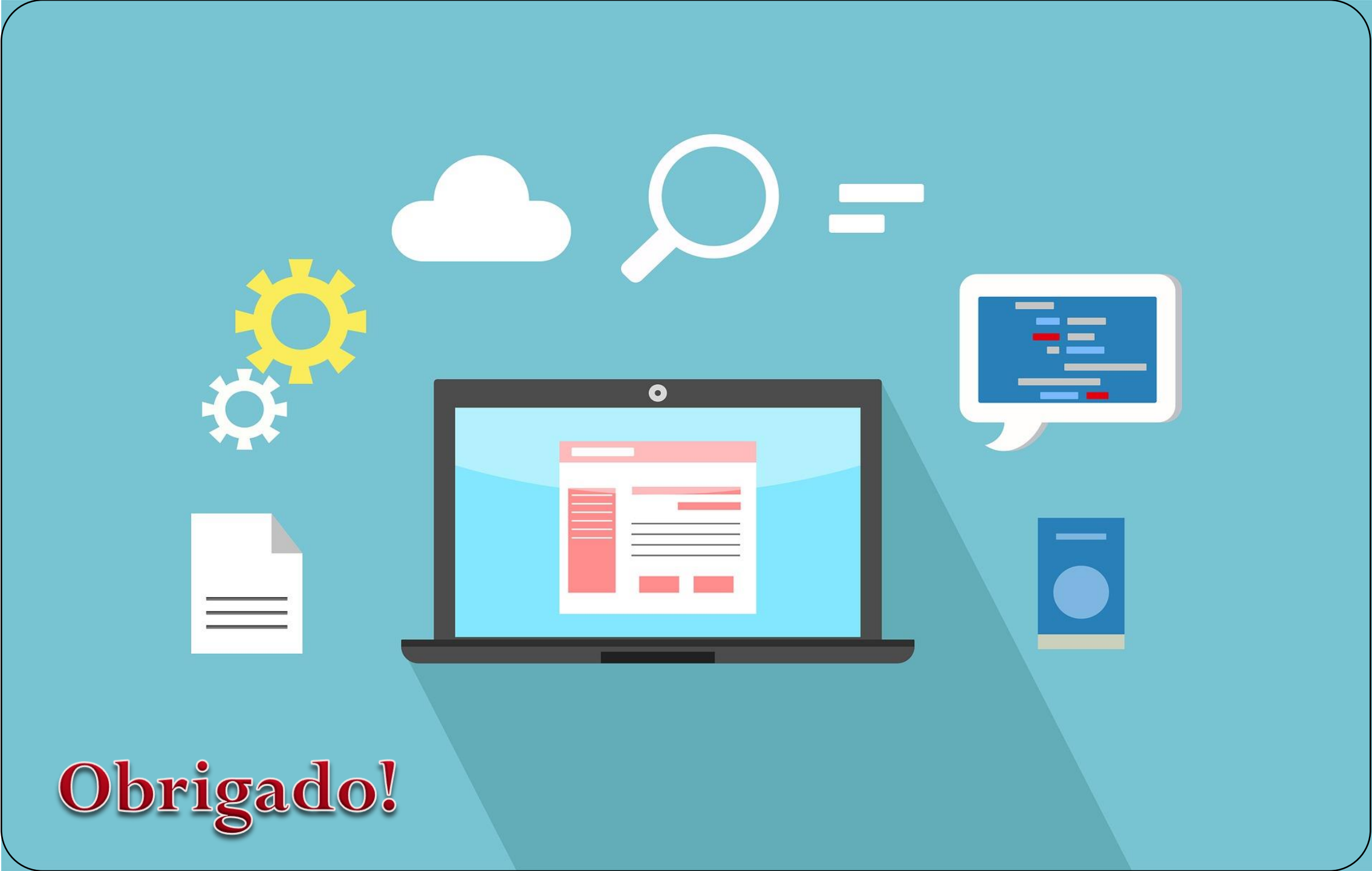
Prática:

1 – Crie uma classe do tipo robô com as características:

- Variáveis de instância:
  - status (do tipo string, pode ser “explorando”, “retornando para casa”);
  - speed (do tipo inteira, pode ser de 1 a 5);
  - temperatura (do tipo real, pode ser de 1 a 1000).
- Métodos:
  - Checar temperatura (se for maior que 660, a velocidade muda para 5 e o status para “retornando para casa”);
  - Mostrar atributos (todo os atributos).

2 – Crie uma classe que utilize essa outra classe (classe main).

3 – Estancie objetos da classe robô e acesse os atributos e métodos.


Obrigado!