

Programação Orientada a Objeto

06. Introdução a POO com Java (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020

Objetivos

- Exercitar os conceitos de POO vistos na aula anterior!
 - Escrevendo, compilando e interpretando (executando) aplicações em Java;
 - Conceitos básicos de POO em Java (classe, objeto, atributos, métodos e mensagens).

Ainda não vamos exercitar entrada de dados (dados serão "hard coded")!

Exercício #1 Parte #0

- Uma empresa de compra e venda de veículos seminovos quer identificar oportunidades de lucro em suas futuras compras. Há uma certa taxa fixa aplicada sobre o valor de um veículo, um imposto, a qual o empresário não quer considerar em suas contas. Um veículo também perde valor agregado de tempos em tempos. Ajude o empresário a modelar uma aplicação que ajuda a entender melhor essas duas questões (incidência de imposto e depreciação).



Exercício #1 Parte #1

- Construir uma classe em Java que representa um veículo, chame-a de **Veiculo**, conforme a especificação a seguir (foque nos atributos, por enquanto):



Veiculo
<code>int ano</code> <code>String marca</code> <code>String modelo</code> <code>double motor</code> <code>double preco</code> <code>char combustivel</code>
<code>void imprimeDados()</code> <code>double calculaImposto()</code> <code>void deprecia(double valor)</code>

Exercício #1 Parte #1

- Construir uma classe em Java que representa um veículo, chame-a de **Veiculo**, conforme a especificação a seguir (foque nos atributos, por enquanto):

```
public class Veiculo{  
  
    int ano;  
    String marca;  
    // outros atributos  
  
}
```

Veiculo
int ano String marca String modelo double motor double preco char combustivel
void imprimeDados() double calculaImposto() void deprecia(double valor)

Exercício #1 Parte #1

- Construir uma classe em Java que representa um veículo, chame-a de **Veiculo**, conforme a especificação a seguir (foque nos atributos, por enquanto):

Considere que os veículos podem ter dois tipos de combustível apenas: gasolina, representado por 'g', e gasolina ou álcool (flex), representado por 'f'!

```
int modelo;  
// outros atributos
```

```
}
```

```
double preco;  
char combustivel
```

```
void imprimeDados()  
double calculaImposto()  
void deprecia(double valor)
```

Exercício #1 Parte #2

- Construir uma classe em Java que representa o programa principal e contém o método *main*. Chame-a de **ProgramaPrincipal**:

```
public class ProgramaPrincipal{  
    public static void main (String[] args){  
        // implementação  
    }  
}
```

Exercício #1 Parte #2

- O método *main* de **ProgramaPrincipal** deve declarar e instanciar um objeto do tipo **Veiculo** e atribuir valores a cada um de seus atributos. Por exemplo:

```
Veiculo v1; // declaração de variável
v1 = new Veiculo(); // instanciação de objeto

v1.ano = 2017;
v1.marca = "GM";
v1.modelo = "Onix";
v1.motor = 1.0;
v1.preco = 42000;
v1.combustivel = 'f';
```


Exercício #1 Parte #2

- O método *main* de **ProgramaPrincipal** deve declarar e instanciar um objeto do tipo **Veiculo** e atribuir valores aos seus atributos. Por exemplo:

Note que se usa o operador ponto (".") para acessar atributos (e métodos) de um objeto instanciado!

```
v1.marca = "GM";  
v1.modelo = "Onix";  
v1.motor = 1.0;  
v1.preco = 42000;  
v1.combustivel = 'f';
```

Exercício #1 Parte #3

- Implementar os métodos de **Veiculo**, conforme a especificação a seguir:

```
public tipoRetorno nomeMetodo (tipoArgumento){  
    // implementação do método  
}
```

- **tipoRetorno** deve ser **void** se não houver retorno ou especificar o tipo de dados de retorno;
- **tipoArgumento** deve ser **void** se não houver argumentos (opcional) ou especificar um tipo de dados. Há um **tipoArgumento** por argumento.

Exercício #1 Parte #3

- Implementar os métodos de **Veiculo**, conforme a especificação a seguir:
- `void imprimeDados()` mostra na tela todos os dados do veículo (use `System.out.println()` para impressão); Formate a saída (Ex.: “Marca: GM”; “Combustível: Flex”, etc.);
- `void deprecia(double valor)` desconta “valor” do atributo “preco”, simulando a depreciação de um veículo;

Veiculo
int ano String marca String modelo double motor double preco char combustivel
void imprimeDados() double calculaImposto() void deprecia(double valor)

Exercício #1 Parte #3

- Implementar os métodos de **Veiculo**, conforme a especificação a seguir:
- **double calculaImposto()** retorna o valor dos impostos já inclusos no valor final do veículo (preço) conforme cilindradas do motor e tipo de combustível:
 - motor até 1.0, gasolina ou flex: 30.6% do valor total;
 - motor até 2.0, gasolina: 36.6% do valor total;
 - motor até 2.0, flex: 34.6% do valor total;
 - motor acima de 2.0, gasolina: 48.6% do valor total;
 - motor acima de 2.0, flex: 41.6% do valor total;

Veiculo
int ano String marca String modelo double motor double preco char combustivel
void imprimeDados() double calculaImposto() void deprecia(double valor)

Exercício #1 Parte #4

- Testar os métodos da instância **v1** já criada em **ProgramaPrincipal**;
 - Imprima os dados de **v1** com o método **imprimeDados**;
 - Deprecie o valor de **v1** em R\$540,00 com o método **deprecia**;
 - Imprima novamente os dados de **v1** com o método **imprimeDados**;
 - Imprima o valor do imposto sobre **v1** ao chamar o método **calculaImposto**.
A impressão do valor do imposto deve ocorrer no método **main** de **ProgramaPrincipal**.

*Ex.: se de **v1.preco** é R\$ 41.460,00 após depreciação, então o cálculo do imposto, assumindo **v1.motor** igual 1.0, deve retornar R\$12.686,76*

Exercício #1 Parte #5

- Crie uma nova instância de **Veiculo**, **v2**, e atribua os mesmos valores que aqueles atribuídos aos atributos de **v1**;
- Compare **v1** e **v2** utilizando o código abaixo:

```
if (v1 == v2)
    System.out.println("Os veículos são iguais");
else
    System.out.println("Os veículos diferem");
```

- O que acontece?

Exercício #1 Parte #6

- Declare uma variável do “tipo” **Veiculo**, **v3**, e atribua **v1** a ela;
- Modifique o valor de **v3**, e imprima os dados de **v1** e **v3**;

Ex.:

```
Veiculo v3 = v1;  
v3.preco = 0; // veículo com perda total  
v3.imprimeDados();  
v1.imprimeDados();
```

- O que acontece?

Atividades

- Entrega de um/dois arquivos **.java**
 - Entregue a implementação do Exercício #1 Partes #1 a #6
 - Indique com um comentário ("//" ou "/**/") o que acontece nos Exercício #1 Partes #5 e #6
 - Entrega até às 23:55h de 18/02/2021



Programação Orientada a Objeto

06. Introdução a POO com Java (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020