TIMTec

Instituto TIM

Introdução à programação em **linguagem JAVA**

AULA 1 | Capítulo 8

Exercício 1

Suponha você está desenvolvendo um aplicativo que precisará conhecer o conceito “Bicicleta”. Quais os atributos e métodos que você poderia incluir na sua classe?

Possível resposta:

```
class Bicicleta{
String cor;
String marca;
String dono;
Int marchas;
Float preco;

void pedalar(){}
void parar(){}
void definirVelocidade(){}
void seEquilibrar(){}
}
```

Exercício 2:

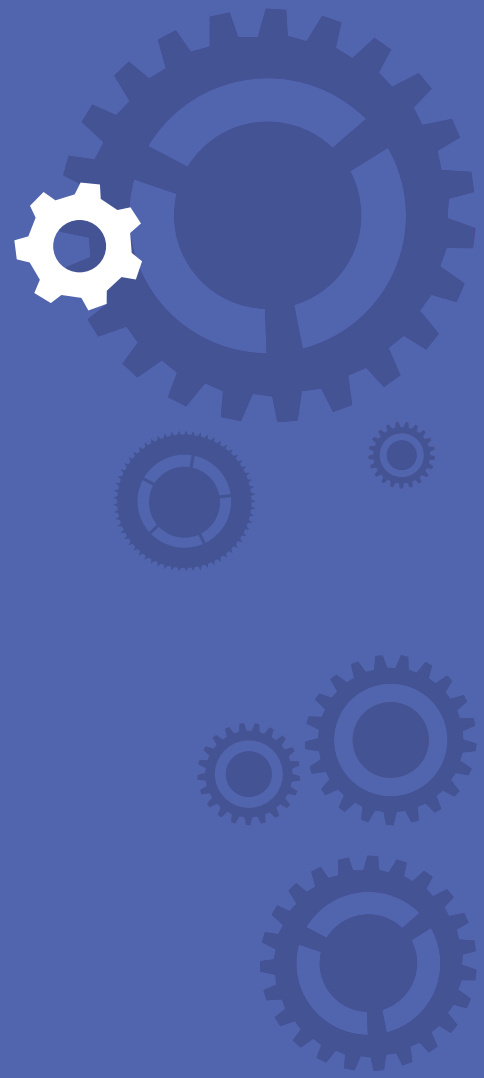
Suponha que o aplicativo que você está desenvolvendo é um jogo de corrida de bicicletas. Os atributos e métodos que você pensou são suficientes? Há propriedades desnecessárias?

Resposta:
Existem diversas maneiras de modelar objetos para uma determinada aplicação. Um jogo de corrida de bicicletas pode precisar de um atributo velocidade e de métodos para acelerar, freiar e virar a bicicleta. Um jogo mais realista poderia ter um atributo para armazenar em que marcha a bicicleta está e esse valor alteria comportamento de outros métodos. Nesse caso é prudente modelar métodos como marchaAcima() e marchaAbaixo().

Se o jogo não possui um sistema de comércio para as bicicletas, a informação de preço, por exemplo, pode ser removida do modelo.

Exercício 3:

Se o aplicativo for para gestão de uma bicicletaria, os métodos e atributos serão os mesmos do jogo? Se não, qual a modelagem que você faria?



Resposta:
Mudar o domínio da aplicação pode levar a mudar consideravelmente a modelagem. Em uma bicicletaria toda bicicleta deverá ter um atributo de código único, para que seja identificada nas ordens de serviço. Ela também deverá ter um outro atributo que identifique o proprietário. Métodos para anexar ordens de serviço e buscar ordens de serviço anexas à bicicleta também poderão ser criados.

Exercício 4:

Explique os seguintes conceitos discutidos em aula:

1.2.a) Classe

1.2.b) Instância

1.2.c) Atributo

1.2.d) Método

Resposta:
http://docs.oracle.com/javase/tutorial/java/concepts/

Exercício 5:

Observe o trecho de pseudocódigo abaixo e explique o resultado esperado.

```
class Pessoa {
String passaporte;
}

rafael = new Pessoa();
rafael.passaporte = "ABC123";
print(rafael.passaporte);
```

Resposta: "ABC123";

```
marcos = new Pessoa();
marcos.passaporte = "XYZ987";
print(marcos.passaporte);
```

Resposta: "XYZ987";

```
marcos.passaporte = rafael.passaporte;
print(marcos.passaporte);
```

Resposta: "ABC123";

AULA 1 | Capítulo 9

Exercício 1:

No capítulo anterior, exercício 1, você fez alguns modelos diferentes para bicicletas. Repita o processo considerando triciclos.

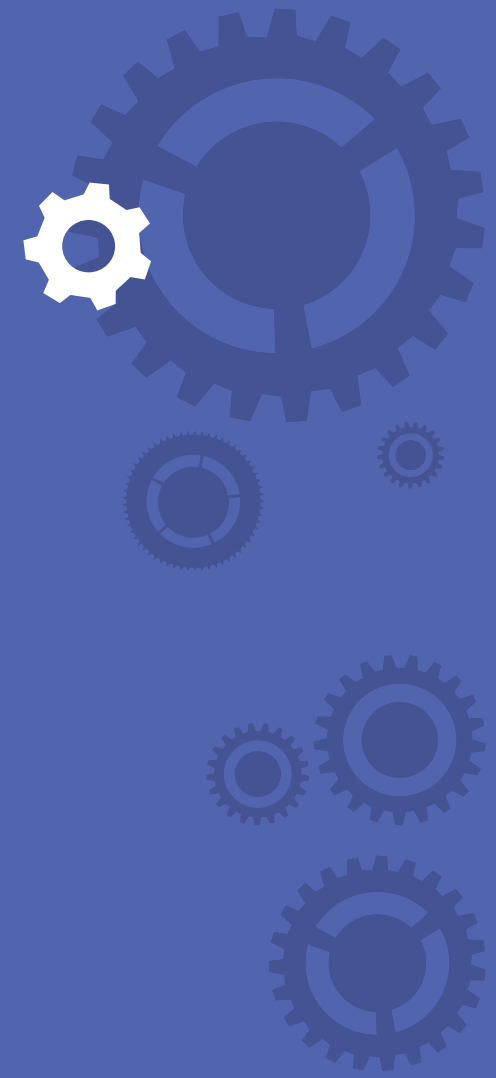
Resposta:

```
class Triciclo{
String cor;
String marca;
String dono;
Int marchas;
Float preco;

void pedalar(){}
void parar(){}
void definirVelocidade()
{}
}
```

Exercício 2:

Identifique os atributos e métodos comuns a bicicletas e triciclos. Crie uma classe veículo e refaça seus modelos de modo que bicicleta e triciclo sejam classes filhas da classe veículo.



Resposta:

```
class Veiculo{
String cor;
String marca;
String dono;
Int marchas;
Float preco;

void pedalar(){}
void parar(){}
void definirVelocidade(){}
}


class Bicicleta extends Veiculo{
int numeroDeRodas = 2;

void seEquilibrar(){}
}

class Triciclo extends Veiculo{
int numeroDeRodas = 3;
}
```

Exercício 3:

Observe o pseudocódigo abaixo:

```
class Pessoa {
void pedalar(Veiculo v){
v.pedalar();
}
}

p = new Pessoa();
b = new Bicicleta();
t = new Triciclo();
```

Quais os requisitos necessários para que "p.pedalar(b);" e "p.pedalar(t); sejam instruções válidas?

Resposta:
É necessário que Bicicleta e Triciclo sejam Veículos, o que pode ser atingido por meio da herança proposta no exercício anterior. É necessário que a classe Veículo defina um método pedalar(), o qual pode ou não ser sobrescrito nas classes filhas.

TIM Tec

Instituto TIM