
Herança

Herança

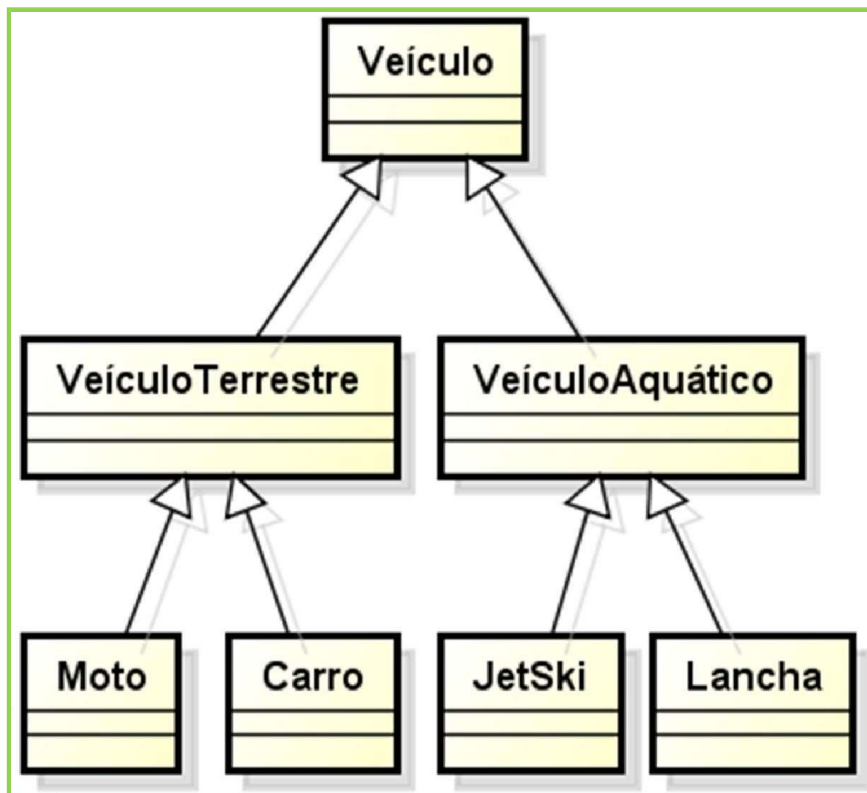
- Herança é uma forma de reutilização de software, onde uma nova classe é criada absorvendo dados de uma classe existente
- Esta nova classe pode ter características mais específicas ou modificadas em comparação com a classe antiga/absorvida
- Com a herança, o tempo de desenvolvimento de um software é reduzido e a depuração é facilitada

Herança

- A classe nova é chamada de SUBCLASSE, já a classe antiga, que é absorvida pela nova, é chamada de SUPERCLASSE
- A herança pode se dar em vários níveis, formando uma hierarquia
- A classe imediatamente superior é uma SUPERCLASSE direta
- Uma classe que não seja imediatamente superior é uma SUPERCLASSE indireta

Herança

- Abaixo um exemplo de herança:



Alguns apontamentos:

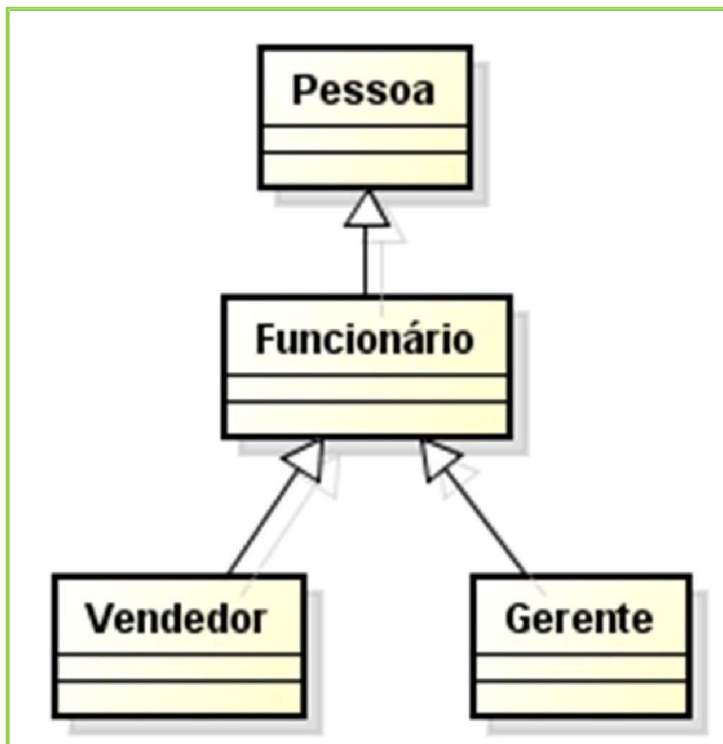
- 1-A Moto é um VeículoTerrestre, mas também é um Veículo
- 2-A Lancha é um Veículo, porém não é VeículoTerrestre
- 3-Moto e JetSki são Veículos, porém um é VeículoTerrestre e o outro é VeículoAquático
- 4-VeículoAquático e VeículoTerrestre compartilham dados da classe veículo

Herança

- Uma subclasse herda os métodos e atributos da sua superclasse, de acordo com os modificadores de acesso estudados
- Desta maneira, uma classe Gerente herda características de Funcionário, que por sua vez herda de Pessoa
- Uma classe Gerente vai ter os dados tanto de Funcionário quanto de Pessoa!

Herança

- Abaixo outro exemplo de herança:

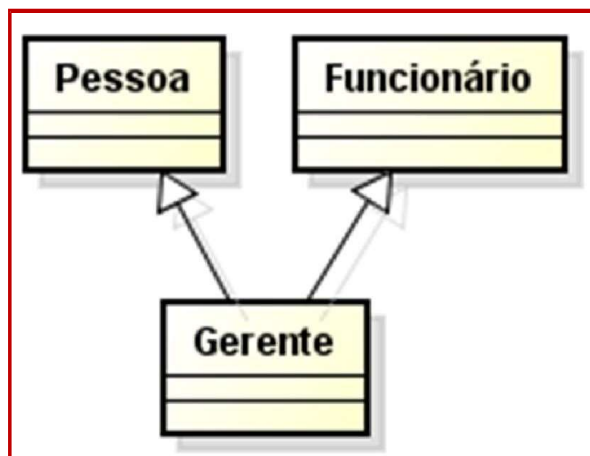


Alguns questionamentos:

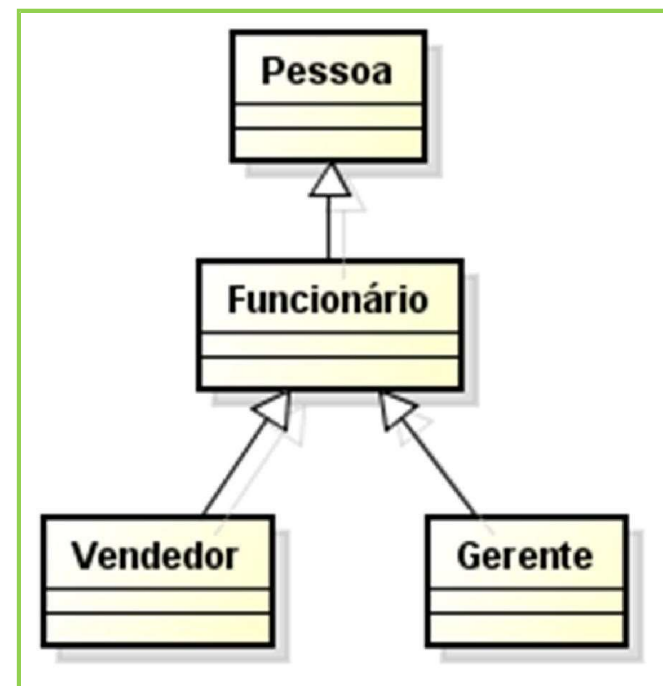
- 1-Poderia ter uma classe acima de Pessoa?
- 2-Poderia ter mais alguma classe abaixo de Vendedor ou de Gerente?
- 3-Poderia ter mais classes ao lado das classes apresentadas no diagrama?
- 4-Todas as características são herdadas ou isso pode ser controlado?

Herança

- Java não suporta herança múltipla:



Assim não pode!



O certo é assim!

7

Herança

- Agora vejamos como ficaria o exemplo dos veículos em código Java:

```
1  import java.util.Date;
2
3  public class Veiculo {
4      private double preco;
5      private Date fabricacao;
6      private double peso;
7  }
```

```
1  public class VeiculoAquatico extends Veiculo {
2      private int velocidadeNautica;
3      private String categoriaNautica;
```

```
1  public class VeiculoTerrestre extends Veiculo{
2      private int velocidadeTerrestre;
3      private String categoriaTerrestre;
4  }
```


Herança

- continuando:

```
1 public class Moto extends VeiculoTerrestre{
2     private String marca;
3     private String proprietario;
4 }
```

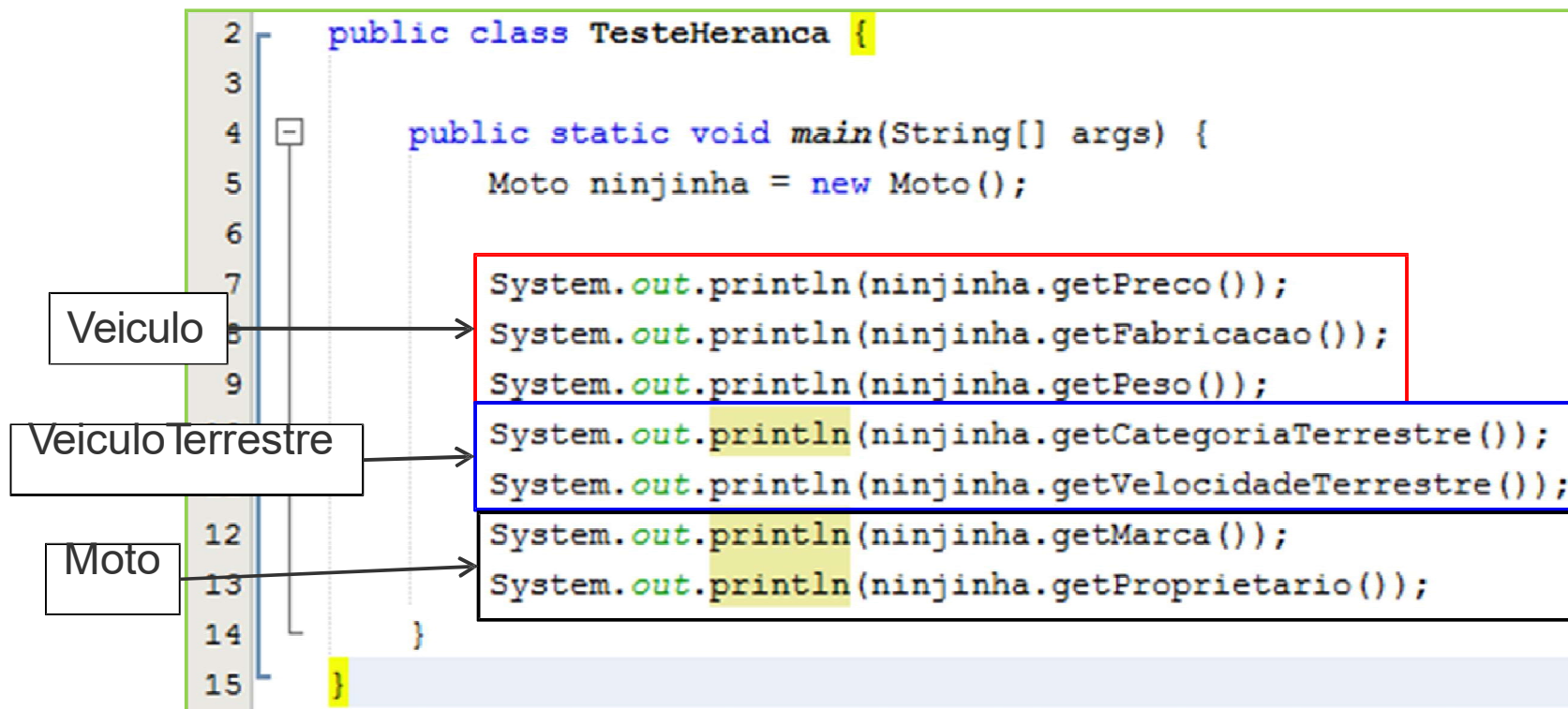
```
1 public class Carro extends VeiculoTerrestre{
2     private String marca;
3     private String proprietario;
4 }
```

```
1 public class JetSki extends VeiculoAquatico{
2     private String marca;
3     private String proprietario;
4 }
```

```
1 public class Lancha extends VeiculoAquatico{
2     private String marca;
3     private String proprietario;
4 }
```

Herança

- Acessando dados da superclasse através das subclasses:



Exercício

- Crie uma hierarquia de herança com 3 níveis: Pessoa, Funcionário e (Vendedor + Gerente)
- Coloque os dados pessoais na classe Pessoa
- Coloque os dados comuns a todos os funcionários na classe Funcionario
- Coloque os dados exclusivos do Gerente na sua classe. O salário do gerente é baseado nas vendas dos vendedores
- Cada Vendedor deve ter um Gerente responsável
- O salário do Vendedor deverá ser por comissão

11



Perguntas?

