



Herança

Ling. Programação Orientada a Objetos
Prof. Leandro Fernandes

1

Herança

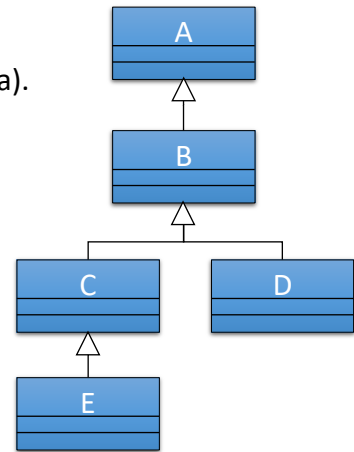
- Permite a criação de uma nova classe a partir de uma classe já existente:
 - absorvendo os dados e comportamentos da classe existente; e
 - aprimorando-a com novas capacidades.
- Portanto, a subclasse estende a superclasse.
 - Uso da palavra reservada **extends**
- Subclasse:
 - Grupo mais especializado de objetos.
 - Comportamentos herdados da superclasse:
 - Podem se personalizar.
 - Comportamentos adicionais.
- Capacidade de reutilização de software.

2

Hierarquia de classes

- Superclasse direta:
 - Herdada explicitamente (um nível acima na hierarquia).
- Superclasse indireta:
 - Herdada de dois ou mais níveis acima na hierarquia.
- Herança única:
 - Herda de apenas uma superclasse (Ex: Java).
- Herança múltipla:
 - Herda de múltiplas superclasses (Ex: C++).

Java não admite/suporta heranças múltiplas



3

Superclasses e subclasses

- Frequentemente, um objeto de uma classe também “é um” objeto de uma outra classe.
 - Exemplo: Em geometria, um *retângulo* é um *quadrilátero*.
 - A classe Retângulo herda da classe Quadrilátero.
 - Quadrilátero: superclasse.
 - Retângulo: subclasse.
- A superclasse geralmente representa um conjunto maior de objetos do que as subclasses.
 - Superclasse: Veículo
 - Carros, caminhões, barcos, bicicletas...
 - Subclasse: Carro
 - Subconjunto mais específico e menor de veículos.

4

Visibilidade e Controle de Acesso

- O modificador de acesso **protected**:
 - Tem um nível intermediário de proteção entre **public** e **private**.
- Os membros **protected** são acessíveis por:
 - membros da superclasse;
 - membros da subclasse; e
 - membros da classe no mesmo pacote.
- É possível, na subclasse, fazer referência a um membro da superclasse usando o **super**.
 - Ele elemento opera de maneira similar ao **this**, porém fazendo referência aos membros hierarquicamente superiores.
- O **super** é utilizado também para informar valores a construtores parametrizados na superclasse.

5

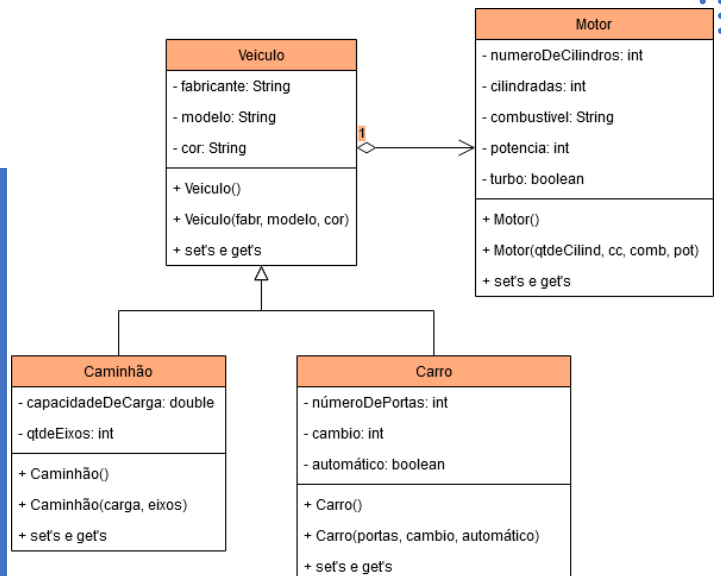
Exemplo: Automobilístico (Hand's on)

Relacionamentos:

- **Agregação:** Veículo -> Motor
- **Herança:** Caminhão -> Veículo
- **Herança:** Carro -> Veículo

Herança vs Associação

- **Herança:** trata de um relacionamento entre classes
- **Associação:** representa relacionamentos entre instâncias de classes.



6

O comando FOR (variante: *for-each*)

- Em Java podemos utilizar o comando FOR para iterar sobre uma lista ou conjunto de elementos.
 - Essa funcionalidade foi introduzida a partir da versão 1.5

- A sintaxe para seu uso é:

```
for(Classe obj : Coleção) {
    //Algum uso de obj
    System.out.println(obj);
}
```

- A cada iteração, o objeto *obj* faz referência a uma das instâncias da coleção.
 - Tal qual o índice de um vetor
- O laço encerra a repetição quando *obj* passar a apontar para *null* (nulo).

7

Aspectos interessantes

- Quando uma declaração de uma classe não estende explicitamente de qualquer outra, o compilador Java configura-a como derivada da classe **Object**.
 - Esta é a classe mais básica de Java, a partir da qual todas as classes de Java são definidas.
- É um erro de sintaxe sobrescrever um método com um modificador de acesso mais restrito.
 - Um método *public* da superclasse não pode se tornar um método *protected* ou *private* na subclasse.
 - Isso quebraria o relacionamento “é um” que exige que todos os objetos de uma subclasse possam responder a chamadas de métodos *public* declarados na superclasse.

8

Exemplo: Sistema Bancário (Hand's on)

- Neste exemplo usamos herança para agrupar os elementos comuns que são compartilhados por todos os tipos existentes de conta bancária.
- Observe que os construtores nas subclasses precisam informar parte de seus parâmetros para a superclasse.
- Note a necessidade de reescrita para os métodos: extrato e depósito.

