

# Interfaces e Classes Abstratas

---

Prof. Dr. Tiago Araújo

[tiagodavi70@ua.pt](mailto:tiagodavi70@ua.pt)

# Interfaces

---

Durante a criação de software, é comum que mais de um grupo de programadores trabalhe no mesmo projeto

É fundamental estabelecer um "contrato" entre os grupos, de forma que os programas possam se comunicar

Não importa como a implementação será feita

- O importante é saber a definição do contrato
- Garante que o software desenvolvido por um grupo se comunica com o outro através deste "contrato"

Em POO, as interfaces fornecem esse contrato

# Interfaces

---

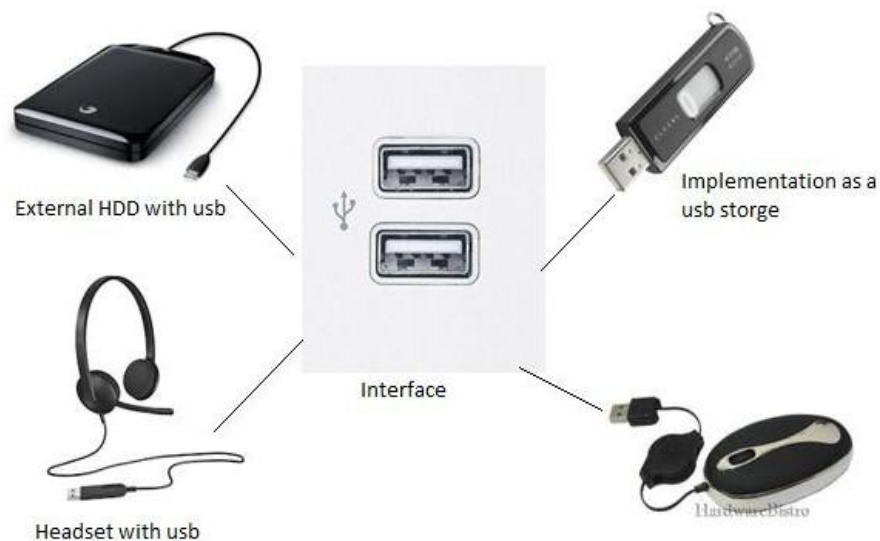


# Interfaces

O padrão USB é um exemplo

Está presente em diferentes dispositivos

As empresas que criam dispositivos que se conectam através de USB, só precisam conhecer o protocolo (mensagens trocadas) de uma conexão USB



# Interfaces

---

Interfaces são um tipo especial de referência, parecido com classes

Normalmente NÃO podem ser instanciadas

Assim como classes, interfaces podem ter níveis de acesso

Podem conter campos constantes

Os métodos são definidos apenas pela sua assinatura

Só podem ser **implementadas** por classes

Podem ser **herdadas** por outra interface

# Interfaces

---

Para usar uma interface, ela deve ser implementada

Todos os métodos precisam ser implementados na classe

# Interfaces

---

Assim como classes, as interfaces podem ser definidas com níveis de acesso

Uma interfaces pode estender (herdar) várias interfaces

Uma classe pode implementar várias interfaces

- Supre a falta de herança múltipla

# Classes Abstratas

---

Em muitos casos, desejamos definir uma classe geral, que representa objetos de maneira genérica, mas que não faz sentido possuir uma instância

Exemplo: Classes Animal, Vaca, Gato, Ovelha

- No mundo real, todo animal é de algum subtipo
- Não faz sentido que exista um objeto Animal
- Porém, a definição da classe Animal é vantajosa, pois permite compartilhar as características comuns de todos os animais

Neste caso, faz mais sentido que a classe Animal seja abstrata

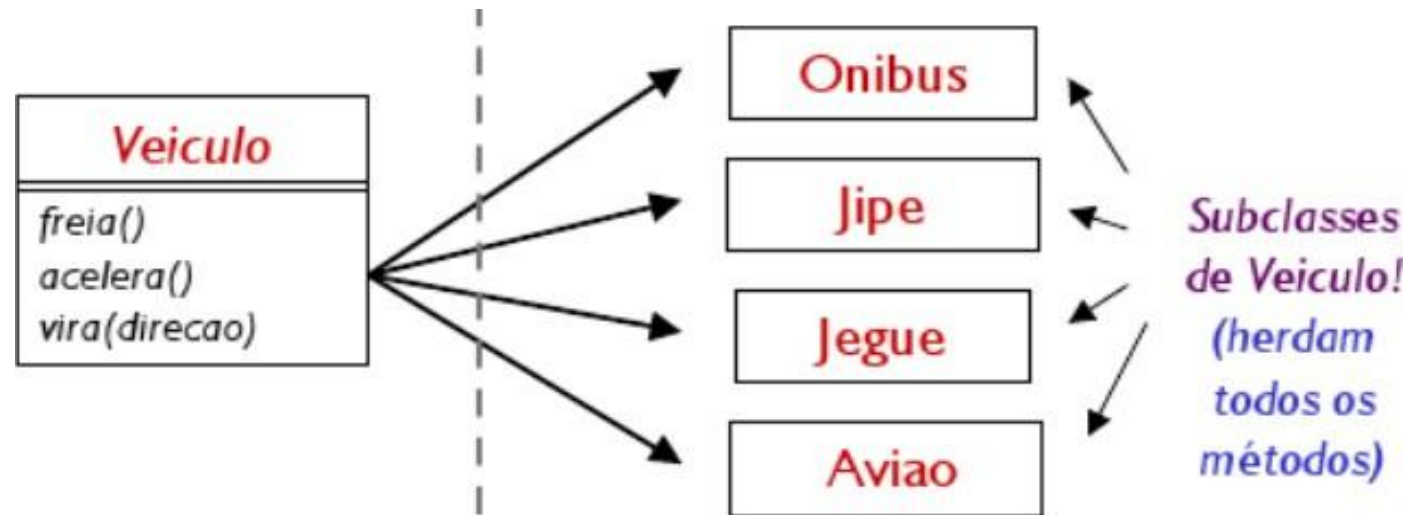
- As outras classes são ditas concretas



# Classes Abstratas

Todo veículo será sempre de um dos subtipos

Definimos, neste problema, que não faz sentido existir instâncias da classe Veiculo



# Classes Abstratas

---

Vimos que os métodos de uma interface são implicitamente abstratos

Não possuem corpo (implementação)

Estabelecem o contrato mas não o comportamento

Obrigam as classes a implementarem

Métodos abstratos também podem ser definidos em uma classe, desde que a classe seja abstrata

# Classes Abstratas

---

## Uma classe abstrata

- Pode conter métodos abstratos e não abstratos
- Pode conter campos como qualquer outra classe
- Não pode ser instanciada
- Pode ser herdada

## Quando classes abstratas são herdadas, métodos abstratos devem ser implementados

- Ou a nova classe deve ser declarada abstrata

# Classes Abstratas

---

Qual utilizar? Depende da aplicação

Em geral, interfaces são utilizadas por classes que não tem relação entre si

- Serializable, Clonable, Comparable
- Não existe uma relação forte (herança) entre as classes

Se há a necessidade de oferecer atributos, interfaces não serão úteis

- Com herança, os atributos serão herdados
- Naturalmente existe uma dependência maior entre as classes

# Sumário

---

	Objetos	Herança	Métodos	Construtor
<b>Interface</b>	Não pode ter instâncias	Uma classe pode <b>implementar</b> várias	Métodos abstratos, default e static	Não pode ter
<b>Classe Abstrata</b>	Não pode ter instâncias	Uma classe pode <b>estender</b> apenas uma	Métodos concretos e abstratos	Pode ter