

Interfaces e Classes Abstratas

Prof. Dr. Tiago Araújo

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

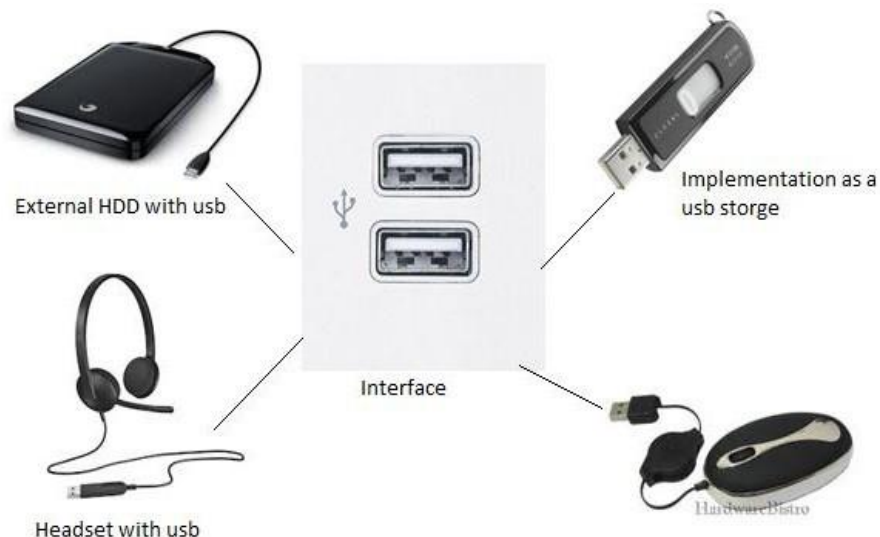


Interfaces

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

Em Kotlin, interfaces são um tipo especial de referência, parecido com classes

NÃO podem ser instanciadas

Assim como classes, interfaces podem ser public ou package-private

Só podem conter campos constantes

- Implicitamente são public, static e final

Os métodos são definidos apenas pela sua assinatura

- Implicitamente são public e abstract

Só podem ser **implementadas** por classes

Podem ser **herdadas** por outra interface

Classes Abstratas

Uma classe abstrata

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

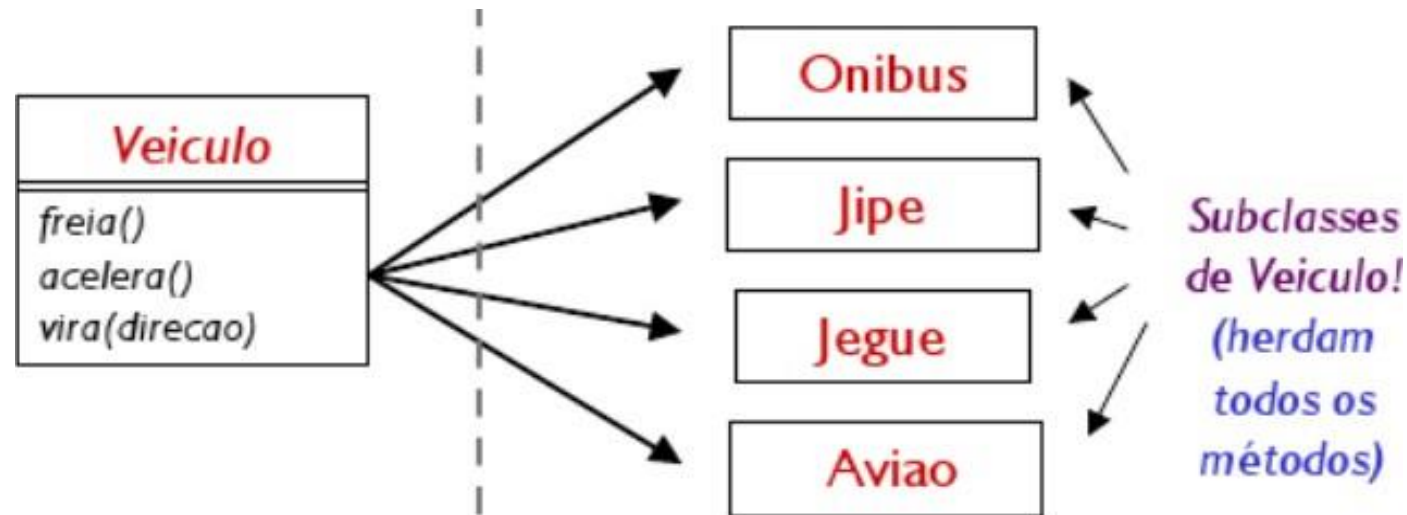
Quando classes abstratas são herdadas, métodos abstratos devem ser implementados

- Ou a nova classe deve ser declarada abstrata

Classes Abstratas

Todo veículo será sempre de um dos subtipos

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



Classes Abstratas

Qual utilizar? Depende da aplicação

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

- 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	Atributos	Construtor
Interface	Não pode ter instâncias	Uma classe pode implementar várias	Métodos abstratos, default e static	Somente constantes	Não pode ter
Classe Abstrata	Não pode ter instâncias	Uma classe pode estender apenas uma	Métodos concretos e abstratos	Constantes e atributos	Pode ter