Java – Aula 15
Interfaces

Cristiano Amaral Maffort

cristiano@cefetmg.br

maffort@gmail.com

Técnico em Informática

Departamento de Computação

CEFET-MG – Belo Horizonte

Introdução

- As classes abstratas, por meio dos métodos abstratos, definem o que uma classe deve fazer
  - Mas sem fornecer uma implementação de como ela fará.
  - O método abstrato fornece apenas a *interface* de um comportamento desejado.
- O objetivo do uso de classes abstratas é definir características “semicompletas” a partir das quais outras classes podem ser construídas para completar as características abstratas.
  - Ou seja, o propósito de uso de classes abstratas e fornecer uma superclasse apropriada por meio da qual outras classes podem herdar *interfaces* e implementações comuns a outras classes.

Definição

- Uma interface é análoga a uma classe abstrata que não contém métodos concretos, apenas métodos públicos e abstratos
  - Ou seja, ela contém apenas a definição das *interfaces* (assinatura dos métodos)
    - Em ambos os casos, as subclasses devem implementar os métodos.
- Em Java, o uso de *interfaces* permite separar totalmente as assinaturas dos métodos da sua implementação
  - Bastando que na definição da classe se utilize a palavra reservada ***interface*** no lugar da palavra ***class***
- Assim, uma *interface* especifica quais operações uma classe (ou uma hierarquia de classes) deve possuir
  - Mas não especifica como elas devem ser implementadas.

Objetivo

- Uma interface é um recurso que permite que o desenvolvedor especifique apenas os ***serviços*** de uma classe
  - As responsabilidades (funcionalidades e/ou comportamento padrão) que uma classe deve possuir.
  - Seus métodos são implicitamente públicos e abstratos.
- Exemplos
  - Aparelho de rádio possui um conjunto de operações padronizadas, tais como mudar de estação, ajustar o volume, escolher entre AM e FM; mas diferentes rádios podem implementar as operações de maneira diferente.
  - Um sistema que necessita de um mecanismo de persistência com operações transacionais básicas (inserir, excluir, alterar, pesquisar).

Objetivo

- A interface estabelece um **contrato** de serviços que devem ser oferecidos por uma classe
  - Ela estabelece as **mensagens** que podem ser trocadas entre componentes de software, **ocultando** completamente os detalhes de implementação.

Interface

- Assim como ocorre com as classes abstratas,
  - Objetos não podem ser instanciados diretamente a partir de uma interface

```
CRUD crud = new CRUD();
```

- Interfaces admitem apenas os níveis de acesso *public* e *default*

- Classes que implementam uma interface devem concretizar todos os métodos dela

Exemplo

```java
package br.cefetmg.inf.persistence;

interface CRUD {
    void inserir(Entity item);
    void excluir(Entity item);
    Entity excluirPorId(Id id);
    void alterar(Entity atual, Entity nova);
    Entity pesquisarPorId(Id id);
    Entity[] pesquisarTodos();
    Entity[] pesquisar(Parameter[] parametros);
}
```

Exemplo

```java
package br.cefetmg.inf.persistence.dao;

import br.cefetmg.inf.persistence.*;

public class AlunoDAO implements CRUD {

    @Override
    public void inserir(Entity item) {  }

    @Override
    public void excluir(Entity item) {  }

    @Override
    public Entity excluirPorId(Id id)  { return null; }

    @Override
    public void alterar(Entity atual, Entity nova)  {  }

    @Override
    public Entity pesquisarPorId(Id id)  { return null; }

    @Override
    public Entity[] pesquisarTodos() { return null; }

    @Override
    public Entity[] pesquisar(Parameter[] parametros) { return null; }
}
```

Interface vs Classe Abstrata

- Uma interface pode ser implementada por várias classes
  - De forma similar ao que ocorre com classes abstratas.
- Java admite **apenas *herança simples* entre classes**
  - Não admite herança múltipla de classes
  - **Mas é possível herança múltipla de interfaces**

```
public class B extends A implements C, D { ... }
```

- Podem ser formadas hierarquias de interfaces

```
public interface K {
    int n=10; void foo(); } // constante estática
public interface J extends K {
    int f=4; void boo(); }
public class L implements J {...}
```

Interface vs Classe Abstrata

- Classes abstratas podem conter métodos não-abstratos
  - Que contém implementação e que podem ser herdados e utilizados por instâncias das subclasses

Bibliografia Obrigatória

- HORSTMANN, Cay S.; CORNELL, Gary. ***Core Java: Fundamentos***. 8. ed. São Paulo: Pearson Prentice Hall, 2020, p. 113 – 120.
- SCHILDT, Herbert; SKRIEN, Dale. ***Programação com Java: uma introdução abrangente***. Porto Alegre: AMGH, 2013, p. 298-320.