Programação Orientada a Objetos

Definição e uso de Interfaces

Prof. Leandro Rodrigues Pinto
<leandrorodp@gmail.com>

Interfaces

Objetos definem sua interação com o mundo exterior através dos **métodos** que eles expõem.

Métodos formam a **interface do objeto** com o mundo **exterior**.

Uma **interface** é um elemento que proporciona uma ligação física ou lógica entre dois sistemas ou partes de um sistema que não poderiam ser conectados diretamente.

Exemplo: botões de uma televisão.

Interfaces

O que é uma interface?

É um recurso utilizado para definir métodos e propriedades de um determinado grupo de classes.

Também chamado de **contrato**.

Os **métodos** definidos no contrato devem ser **implementados** pela classe que selar o contrato.

Interfaces

- **Interfaces** aceitam apenas **assinaturas** de **métodos** e propriedades.

- O comportamento concreto será **implementado** pelas **classes**.

- **Uma classe** pode selar contrato com **mais de uma interface**.

- **Uma interface** pode ser implementada por **mais de uma classe**.

Implementação de Interface

A implementação de uma **interface** permite que uma classe se torne mais formal sobre o comportamento que promete fornecer.

As **interfaces** formam um **contrato** entre **a classe e o mundo externo**, e esse contrato é imposto pelo **compilador** em tempo de construção.

Se sua classe implementa uma interface, **todos** os métodos definidos por essa interface deverão aparecer em seu código-fonte antes que a classe seja compilada com êxito.

Implementação de Interface

```
public interface Calculadora {

    // especificações dos métodos
    public double somar(double x, double y);
    public double subtrair(double x, double y);
    public double multiplicar(double x, double y);
    public double dividir(double x, double y);

}
```

- Veja que existe a palavra reservada "**interface**" no lugar de "**class**".

- O nome do arquivo será o mesmo da interface, da mesma maneira que ocorre com as classes: Calculadora.java.

Implementação de Interface

Lembre-se:

extends = herança

implements = interface

```
public interface Calculadora {

    // especificações dos métodos
    public double somar(double x, double y);
    public double subtrair(double x, double y);
    public double multiplicar(double x, double y);
    public double dividir(double x, double y);

}
```

```
public class Smartphone implements Calculadora {

    @Override
    public double somar(double x, double y) {
        return x + y;
    }
    // Implementação dos métodos contratados

}
```

Interfaces - Resumo

- **Interfaces**, assim como as classes abstratas, **não podem** ser utilizadas para **criar objetos**.
- Ela só expõe o que **o objeto deve fazer**, e não **como ele faz**, nem **o que ele tem**.
- **Como ele faz** vai ser definido em uma **implementação** dessa interface.
- Métodos de interfaces **não** possuem corpo, quem o fornece é a classe que irá implementar a interface.
- Ao implementar uma interface a classe deve sobrecarregar(**@Override**) todos os métodos contratados.
- **Métodos** de interfaces são por padrão **abstract** e **public**.
- **Atributos** de interface são por padrão **public**, **static** e **final**.
- Uma **interface não** pode ter um método **construtor**.

Atividade prática

1º - Criar o diagrama de classes

2º - Implementar

3º - Testar

Exercício

Crie uma interface chamada “Jogo” que especifica os métodos “andar”, “pular” e “chutar”.

A seguir, crie uma classe chamada “Jogar” que implementa esses métodos.

Dentro de cada método imprima uma mensagem em tela contendo a ação correspondente.

Para testar, elabora uma terceira classe chamada “TestarJogo”.

Bibliografia

Cadenhead, Rogers; LEMAY, Laura, Aprenda em 21 dias Java 2. 4. ed. São Paulo: Campus, 2005.

DEITEL, Harvey H .; DEITEL, Paul J. Java: como Programar. 8. Ed. São Paulo: Pearson / Prentice-Hall, 2010.

Perguntas?

Obrigado!