Programação orientada a objetos em Java

Interface

Profª. Heloisa Moura

Programação orientada a objetos em Java
Interface

As interfaces são padrões definidos através de contratos ou especificações. Um contrato define um determinado conjunto de métodos que serão implementados nas classes que assinarem esse contrato. Uma interface é 100% abstrata, ou seja, os seus métodos são definidos como abstract, e os atributos(variáveis) por padrão são sempre constantes (static final).

Uma interface é definida através da palavra reservada “interface”. Para uma classe implementar uma interface é usada a palavra “implements”,

Programação orientada a objetos em Java
Interface

Como a linguagem Java não tem herança múltipla, as interfaces ajudam nessa questão, pois uma classe pode ser herdada apenas uma vez e pode implementar inúmeras interfaces.
As classes que forem implementar uma interface terão de adicionar todos os métodos da interface ou se transformar em uma classe abstrata.
Dentro das interfaces existem somente assinaturas de métodos e pode ter atributos que são constantes. A classe que a utilizará deve realizar a implementação das assinaturas, dando comportamentos práticos aos métodos.

Programação orientada a objetos em Java
Interface

Vamos ver um exemplo de uma interface chamada FiguraGeometrica com três assinaturas de métodos que virão a ser implementados pelas classes referentes às figuras geométricas.

Interface FiguraGeometrica

```
public interface FiguraGeometrica{
        public String getNomeFigura();
        public int getArea();
        public int getPerimetro();
}
```

Programação orientada a objetos em Java
Interface

Para realizar a chamada/referência a uma interface por uma determinada classe, é necessário adicionar a palavra-chave implements ao final da assinatura da classe que irá implementar a interface escolhida.

Sintaxe:

public class nome_classe implements nome_interface

Onde:

nome_classe – Nome da classe a ser implementada.

nome_Interface – Nome da interface a se implementada pela classe.

Programação orientada a objetos em Java
Interface

Exemplo pratico
No Eclipse vamos ver o exemplo de duas classes que implementam a interface FiguraGeometrica, uma chamada Quadrado e outra Triangulo.

```
public class Quadrado implements FiguraGeometrica {
                        .......
}
public class Triangulo implements FiguraGeometrica {
                        .........
}
```

Programação orientada a objetos em Java
Interface

Como foi possível ver, ambas as classes seguiram o contrato da interface FiguraGeometrica, porém cada uma delas a implementou de maneira diferente.

Ao contrário da herança que limita uma classe a herdar somente uma classe pai por vez, é possível que uma classe implemente varias interfaces ao mesmo tempo.

Imagine, por exemplo, uma interface chamada Veiculo e outra chamada Motor.

Programação orientada a objetos em Java
Interface

Interface Veiculo

```
public interface Veiculo {
        .........
}
```

Interface Motor

```
public interface Motor {
        .......
}
```

Programação orientada a objetos em Java
Interface

A seguir é possível visualizar a implementação das interfaces em uma classe chamada Carro.

```
public class Carro implements Veiculo,Motor{
.....
}
```

Programação orientada a objetos em Java
Interface

Interfaces versus classes abstratas

Uma interface costuma ser utilizada no lugar de uma classe abstract quando não há nenhuma implementação padrão a herdar — isto é, nenhum campo e nenhuma implementação padrão de método. Como as classes public abstract, interfaces são tipicamente tipos public. Assim como uma classe public, uma interface public deve ser declarada em um arquivo com o mesmo nome que o da interface e a extensão de arquivo .java.

Conclusão

Por fim, interface nada mais que uma espécie de contrato de regras que uma classes deve seguir em um determinado contexto. Como em Java não existe herança múltipla, a interface passa a ser uma alternativa.

Quadro comparativo entre Interface X Classe abstrata

Interface	Classe Abstrata
Herança múltipla permitido; uma interface pode estender várias interfaces	Herança múltipla não é possível; uma classe só pode estender uma única classe
palavra chave **implements** é utilizada para implementar uma interface	palavra chave **extends** é utilizada para estender uma classe
por padrão todos os métodos são públicos e abstratos (public abstract) - não tem necessidade de declarar os mesmos	métodos podem ter modificares public e abstract se necessário, e podem utilizar outros modificares também
interfaces não tem implementação	podem ter implementação parcial
todos os métodos de uma interface precisam ser sobrescritos	somente métodos abstratos precisam ser sobrescritos (obrigatório)
todas as variáveis declaradas numa interface são **public static final** (constantes)	variáveis podem ser declaradas como **public static final** se necessário, mas não é obrigatório
interfaces não tem construtor(es)	classes abstratas podem ter construtores
métodos não podem ser estáticos (**static**)	métodos não abstratos podem ser estáticos (**static**)

Programação orientada a objetos em Java
Interface

Exercício