

Interface



Prof. Bruno Gomes
bruno.gomes@ifrn.edu.br

Programação Orientada a Objetos

Abstração (Revisão)

- Decompor um sistema complicado em suas partes fundamentais
- Descrevê-las em uma linguagem simples e precisa
 - Atribuir-lhes um nome e descrever suas funcionalidades
- Abstrair os detalhes desnecessários
- Modelar conceitos, entidades, elementos, problemas e características do mundo real levando em conta apenas os detalhes importantes
- Vantagem:
 - Robustez

Abstração - Necessidades

- Definir um conjunto de requisitos para as classes que precisam adequar-se a ela
- É necessário descrever o que as classes devem fazer, sem especificar como devem fazer
- Tipos Abstratos de Dados (TADs):
 - Define o que cada operação faz, mas não como o faz
- Em Java:
 - Expresso por uma interface

Interface

- Visão externa da classe
 - Membros acessíveis pelo usuário da classe
 - Conjunto de métodos públicos
 - Pode ter constantes
 - atributos final
 - Ao se usar um objeto de uma classe, o programador conhece a interface do objeto
 - Uma interface define um contrato a ser cumprido
 - Podemos definir apenas interface

Interface

- Sintaxe:

```
public interface NomeInterface {  
  
    // Assinaturas dos métodos;  
  
}
```

Interface

- Exemplo:

```
public interface Conta {  
    void depositar(double valor);  
    void sacar (double valor);  
    double getSaldo();  
}
```

Definição de Interface

- Similar a definição de classe
- Todos os métodos são:
 - Públicos
 - Abstratos
- Todos os atributos são:
 - públicos
 - estáticos
 - finais
- Uma interface pode estender outras interfaces
 - usando extends
 - `public UmaInterface extends Outra1, Outra2 {...}`

Definição de Interface

- Não pode:
 - Ter Campos de instância
 - Implementar Métodos

Implementação de Interface

- Classes podem implementar uma interface
 - Usando **implements**
 - Classe deve implementar TODOS os métodos definidos na interface

Implementação de Interface

Classe
implementa
Interface

```
public interface Conta {  
    void depositar(double valor);  
    void sacar (double valor);  
    double getSaldo();  
}
```

```
class ContaImpl implements Conta{  
    public void depositar(double valor) {  
        // ...  
    }  
    public double getSaldo() {  
        // ...  
        return 0;  
    }  
    public void sacar(double valor) {  
        // ...  
    }  
}
```

Métodos
definidos
na interface

Implementação de Interface

- Uma classe pode implementar várias interfaces
 - `class Classe implements I1, I2, I3 {...}`
- Se uma classe não implementar todos os métodos de uma interface deve ser abstrata
- Ao implementar uma interface a classe está assinando um contrato definido
- A programação deve ser voltada para interfaces
 - Parâmetros de métodos
 - Referências a objetos

Interfaces como tipo

- Interfaces são tipos
 - Podem ser declaradas variáveis para interfaces
 - Apenas os métodos da interfaces são acessados
 - Uma classe pode ter várias interfaces
 - Vários tipos
 - Somente instâncias de classes que implementam uma determinada interface podem ser referenciadas por uma variável que referencia esta interface
 - Pode utilizar *instanceof* para verificar se um objeto implementa uma interface

Interfaces como tipo

```
public static void main(String args[]){  
    Conta conta;  
    conta = new ContaImpl();  
    double valor=0;  
    /*...*/  
    conta.depositar(valor);  
}
```

Interfaces

```
public void inserir(Contato contato){  
    contatos[qtd++] = contato;  
}
```

Exercício

- Implemente o seguinte diagrama:

