



UNIVERSIDADE
ESTADUAL DE LONDRINA

Interfaces

Laboratório de Programação (5COP011)

Prof. Bruno Bogaz Zarpelão

Interface

- O conceito de interface está relacionado a padronização;
- Na orientação a objetos, os objetos interagem por meio de troca de mensagens;
- Esta troca de mensagens é baseada na invocação de métodos (operações);
- Para facilitar a troca de mensagens, podemos padronizar os métodos oferecidos por um objeto;

Interface

- A construção de uma interface é similar a definição de um contrato;
- A interface é um contrato que determina quais métodos devem ser implementados pelas classes que assinarem este contrato;
- A interface contém apenas a assinatura dos métodos;
- As classes que implementam as interfaces trazem a implementação destes métodos;

Interface

```
public interface ContaBancaria{  
    void sacar(BigDecimal valor);  
}
```

Interface

```
public class ContaCorrente implements ContaBancaria{  
  
    void sacar(BigDecimal valor){  
        BigDecimal saldoTotal = this.saldo.add(this.limite);  
        if (valor.compareTo(saldoTotal) <= 0){  
            this.saldo = this.saldo.subtract(valor);  
        }  
    }  
}
```

Interface

```
public class ContaPoupanca implements ContaBancaria{  
  
    void sacar(BigDecimal valor){  
        if (valor.compareTo(this.saldo) <= 0){  
            this.saldo = this.saldo.subtract(valor);  
        }  
    }  
}
```

Interface

```
public class Principal{  
  
    public static void main(String args[]){  
        ContaBancaria cc = new ContaCorrente();  
        ContaBancaria cp = new ContaPoupanca();  
        cc.sacar(new BigDecimal(100));  
        cp.sacar(new BigDecimal(100));  
    }  
}
```

Duas implementações diferentes!

Interface

```
public class AgenciaBancaria {  
    public void gravarConta(ContaBancaria cb){  
        ... //implementação  
    }  
}
```

Podemos passar como parâmetro tanto objetos do tipo ContaPoupanca como objetos do tipo ContaCorrente, pois eles implementam ContaBancaria.

Interface

- Não é preciso colocar modificador de acesso na assinatura do método na interface;
- Um método em uma interface é automaticamente *public*;
- Uma interface não tem atributos de instância, mas pode ter atributos de classe (*static*) constantes (*final*);
- Todo atributo declarado em uma interface é automaticamente definido como *static* e *final*, independente de sua declaração.

Interface

- Uma classe pode implementar várias interfaces;

```
public class ContaPoupanca implements ContaBancaria, Serializable{  
  
    void sacar(BigDecimal valor){  
        if (valor.compareTo(this.saldo) <= 0){  
            this.saldo = this.saldo.subtract(valor);  
        }  
    }  
}
```

Interface

- Uma interface pode estender outra interface:

```
public interface RepresentacaoGrafica{  
    void desenhar();  
}
```

```
public interface RepresentacaoGraficaRetangular  
                extends RepresentacaoGrafica{  
    double obterAreaRetangular();  
}
```

Interface

```
public class ImplementacaoRepGrafica
    implements RepresentacaoGraficaRetangular{

    void desenhar(){
        ... //implementação
    }

    double obterAreaRetangular(){
        ... //implementação
    }
}
```

Exercício

Observe a interface abaixo e responda as questões a seguir.

```
public interface FormaGeometrica {  
    public int obterNumeroLados();  
}
```

- a) Escreva o código de uma classe Quadrado que implemente FormaGeometrica.
- b) Escreva um trecho para um programa principal que crie um objeto do tipo FormaGeometrica (e seja instância da classe Quadrado) e invoque o método obterNumeroLados().