

Interface Java

Aula 13 – parte 1

Prof. Grace



Interfaces

- Definem e padronizam comportamentos
- Usadas para classes não relacionadas mas com alguns comportamentos semelhantes
- Ex.: Métodos das classes Carro e Pessoa em um jogo
 - andarParaFrente()
 - virarParaDireita()
 - virarParaEsquerda()
 - parar()
- Como resolver? Criar uma herança faz sentido?
- Usamos uma interface que defina esse comportamento!

Interfaces

- Vantagens:
 - Padronização no acesso aos métodos
 - Usadas para implementar herança múltipla
- Exemplos de uso:
 - Acesso a BD
 - Interfaces gráficas de Usuário (GUI)

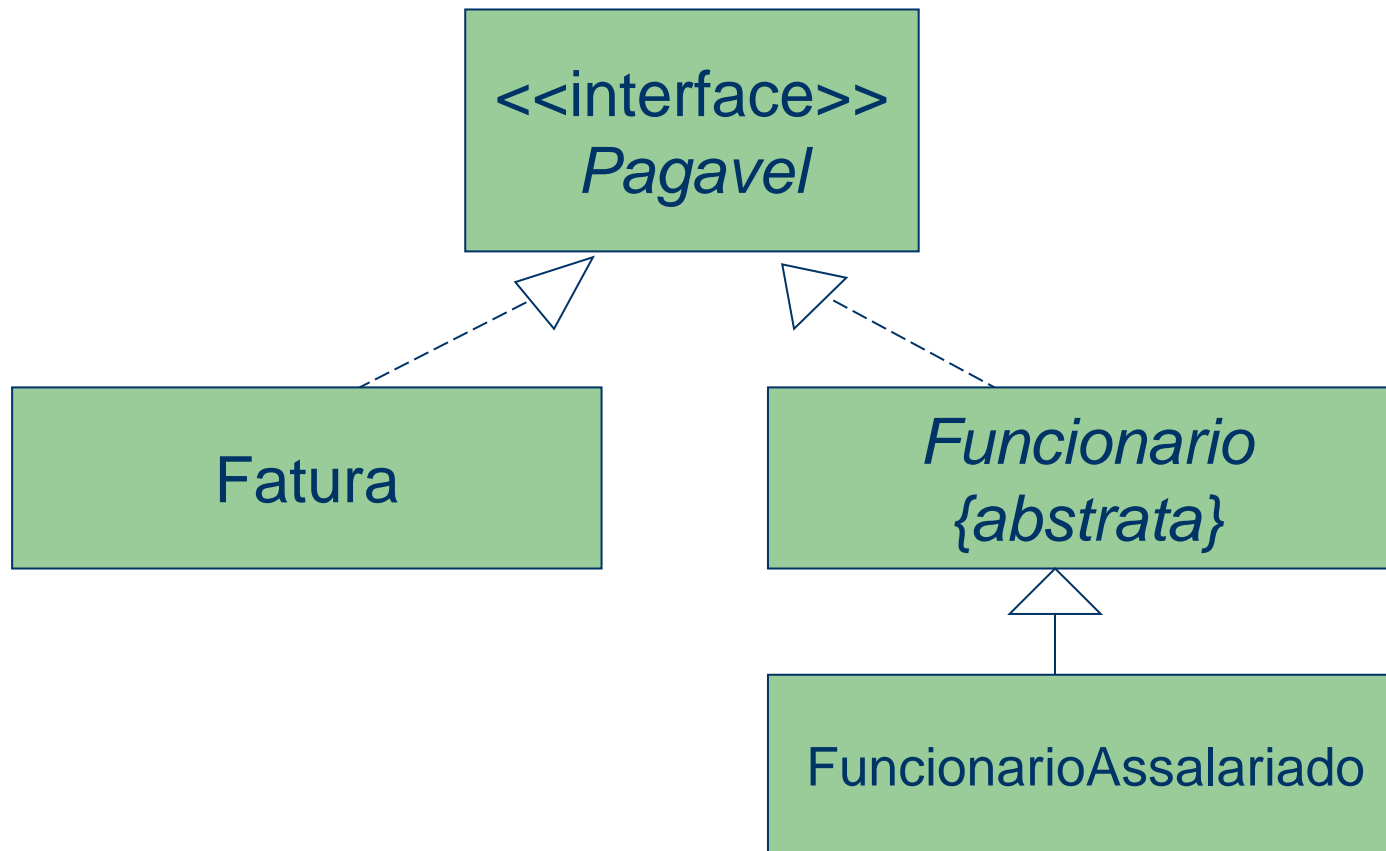
Interface Java

- Descreve conjunto de métodos para instruir como um objeto deve se comportar
- Possui apenas **métodos abstratos**
- Todos os membros devem ser **public**
- Não pode ter nenhum método implementado

Implementação de interface

- Uma classe concreta pode especificar que **implementa** (*implements*) uma ou mais interfaces;
- Nesse caso, todos os métodos devem ser codificados;
- Caso algum método não seja implementado, essa classe deve ser abstrata;
- Implementar uma interface é como assinar um contrato!

Exemplo: Hierarquia usando interface



Especificação

- **Interface Pagavel**

`getPagamento(): double; //não implementa o código`

- **Classe Fatura // classe concreta**

- Atributos: qtd; preco;

- Métodos:

- `<<construtor>> (qtd, preco)`

- `getQuantidade(): integer;`

- `getPreco(): double;`

- `getPagamento(): double; //implementa o código`

Especificação

- **Classe Funcionario {abstrata}**
 - Atributos: matricula; nome; cargo;
 - Métodos:
 - <<construtor>> (matricula, nome, cargo);
 - setCargo (nomeCargo);
 - getCargo(): String;
 - exibeFuncionario();
 - getPagamento(); {método abstrato}***

Especificação

- **Classe FuncionarioAssalariado // concreta**
 - Atributos: salario;
 - Métodos:
 - <<construtor>> (matricula, nome, cargo, salario);
 - setSalario (vlr);
 - getPagamento(): double;**
 - exibeFuncionario();

Implementação

```
//Pagavel.java  
public interface Pagavel{  
    public double getPagamento( );  
}
```

//Fatura.java

```
public class Fatura implements Pagavel{
    private int qtd;
    private double preco;
    public Fatura(int qtd, double preco){
        if (qtd < 0)
            System.out.println("Quantidade de itens deve ser maior que zero");
        else
            this.qtd = qtd;

        if (preco < 0)
            System.out.println("O preco dos itens deve ser maior que zero");
        else
            this.preco = preco;
    }
    public int getQuantidade( ){
        return qtd;
    }
    public double getPreco( ){
        return preco;
    }
    public double getPagamento( ){
        return qtd*preco;
    }
}
```

```
//Funcionario.java
public abstract class Funcionario implements Pagavel{
    int matricula;
    String nome, cargo;

    public Funcionario(int matricula, String nome, String cargo){
        this.matricula = matricula;
        this.nome = nome;
        setCargo(cargo);
    }
    public void setCargo (String nomeCargo){
        this.cargo = cargo;
    }
    public String getCargo( ){
        return this.cargo;
    }
    public void exhibeFuncionario( ){
        System.out.println(nome);
        System.out.println(cargo);
        System.out.println(getPagamento());
    }
    public abstract double getPagamento( );
}
```

```
//FuncionarioAssalariado.java
public class FuncionarioAssalariado extends Funcionario{
    double salario;

    public FuncionarioAssalariado (int matricula, String nome, String cargo, double salario){
        super(matricula, nome, cargo);
        setSalario(salario);
    }
    public void setSalario(double salario){
        if (salario < 0)
            System.out.println("Salario não pode ser negativo!");
        else
            this.salario = salario;
    }
    public double getPagamento( ){
        return salario;
    }
    public void exibeFuncionario( ){
        super.exibeFuncionario();
        System.out.println(salario);
    }
}
```

Programa de teste

```
//SistemaFinanceiro.java
import java.util.Scanner;
public class SistemaFinanceiro{
    public static void main (String args[]){
        int qtdItens, matricula;
        double salarioMensal, precoItens;
        String tipo, nome, cargo;
        Pagavel p;
        Scanner sInput = new Scanner(System.in);
        Scanner nInput = new Scanner(System.in);
        System.out.println("Tipo de pagamento? (f - fatura/ s -salario)");
        tipo = sInput.nextLine();
        if(!tipo.equalsIgnoreCase("f") && !tipo.equalsIgnoreCase("s"))
            System.out.println("Tipo de pagamento não identificado!");
        else{
```

...

```

        if(tipo.equalsIgnoreCase("f")){
            System.out.println("Quantidade de itens: ");
            qtdItens = nInput.nextInt();
            System.out.println("Preço por item: ");
            precoItens = nInput.nextDouble();
            p = new Fatura(qtdItens, precoItens);
        }
        else{
            System.out.println("Matricula: ");
            matricula = nInput.nextInt();
            System.out.println("Nome: ");
            nome = sInput.nextLine();
            System.out.println("Cargo: ");
            cargo = sInput.nextLine();
            System.out.println("Salário: ");
            salarioMensal = nInput.nextDouble();
            p = new FuncionarioAssalariado(matricula, nome, cargo, salarioMensal);
        }
        System.out.println(p.getPagamento());
    }
} // fim do main
} // fim da classe

```