

Polimorfismo com interfaces

Padrões de Projeto Orientado a
Objetos

Profa. Danielle Martin

Prof. Pedro Toledo

Universidade de Mogi das Cruzes

Polimorfismo

Polimorfismo é a característica da orientação a objetos que desacopla a **referência** de um objeto da **implementação** do mesmo, permitindo que um **tipo comum** (abstrato ou genérico) possa utilizar um objeto de qualquer classe polimórfica sem diferenciação.

Exemplo:

```
Animal objeto = new Cachorro();
```

Variavel de referencia
do tipo Animal

Objeto instancia do tipo
Cachorro

Polimorfismo

É possível implementar o polimorfismo usando:

- Herança
- Classes abstratas
- Interfaces

III. INTERFACES

Interface

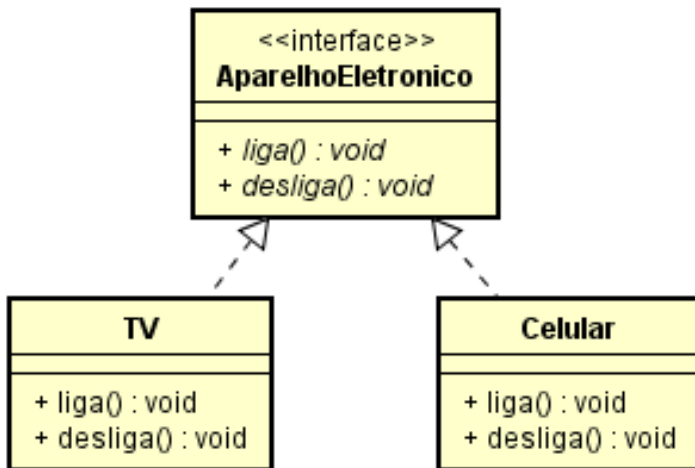
Interface é uma espécie de “contrato” de um comportamento que uma ou mais classes devem implementar.

O contrato especifica **o que** vai ser implementado, mas não especifica **como** – a interface é 100% abstrata.

Uma classe pode implementar uma interface usando a palavra reservada **implements**.

Exemplo

```
public interface AparelhoEletronico {  
    public void liga();  
    public void desliga();  
}
```



```
public class TV implements AparelhoEletronico{  
  
    public void liga() {  
        System.out.println("Liga TV");  
    }  
    public void desliga() {  
        System.out.println("Desliga TV");  
    }  
}
```

```
public class Celular implements  
AparelhoEletronico{  
  
    public void liga() {  
        System.out.println("Liga celular");  
    }  
    public void desliga() {  
        System.out.println("Desliga celular");  
    }  
}
```

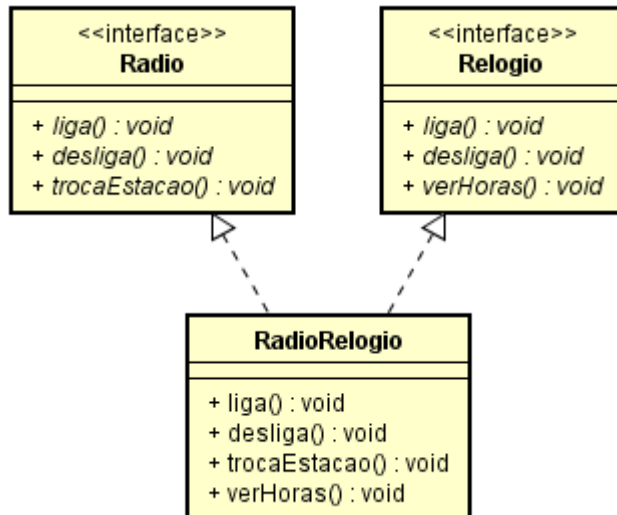
Características de interfaces

- Uma interface não pode conter métodos concretos, seus métodos são implicitamente públicos e abstratos.
- Uma interface não pode conter métodos estáticos.
- Uma interface não pode ter construtores, pois não é possível instanciar objetos de interfaces.
- Todos os atributos declarados em uma interface são implicitamente constantes (`public static final`).
- Uma classe pode implementar mais de uma interface.

Classe implementando mais de uma interface

```
public interface Radio {  
    public void liga();  
    public void desliga();  
    public void trocaEstacao();  
}
```

```
public interface Relogio {  
    public void liga();  
    public void desliga();  
    public void verHoras();  
}
```



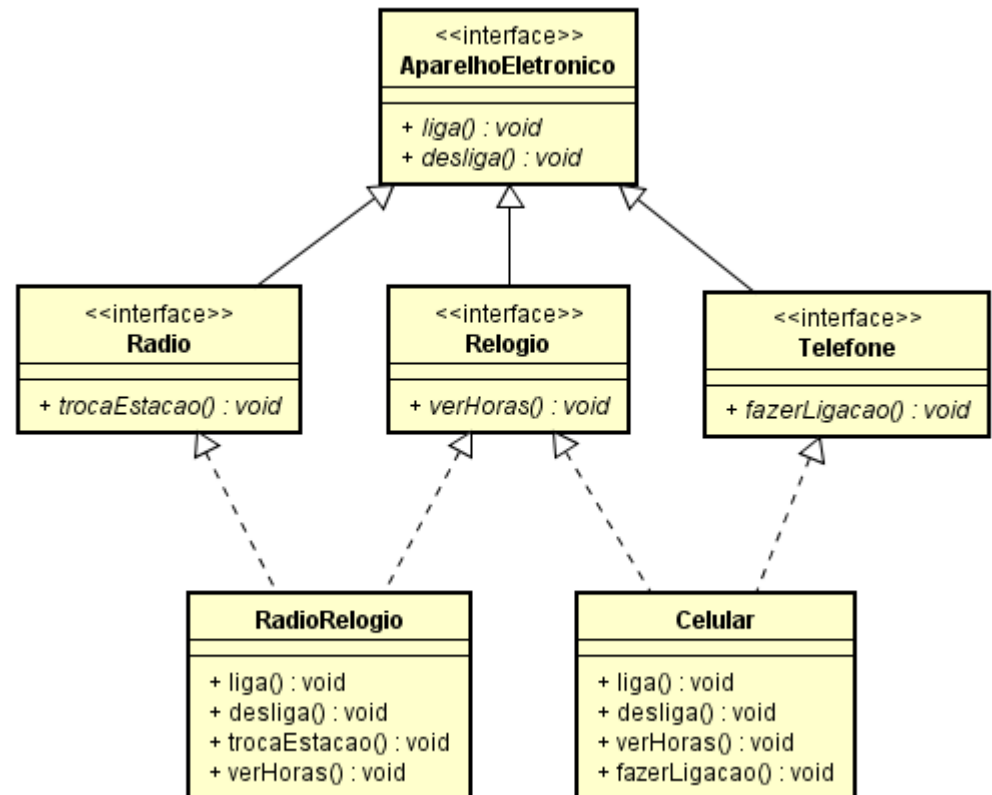
```
public class RadioReligio implements Radio, Relogio  
{  
  
    public void liga() {  
        System.out.println("Liga Radio Relogio");  
    }  
  
    public void desliga() {  
        System.out.println("Desliga Radio Relogio");  
    }  
  
    public void trocaEstacao() {  
        System.out.println("Trocando estacao");  
    }  
  
    public void verHoras() {  
        System.out.println("Agora são 20:00");  
    }  
  
}
```


Herança entre interfaces

```
public interface AparelhoEletronico
{
    public void liga();
    public void desliga();
}
```

```
public interface Radio extends
AparelhoEletronico
{
    public void trocaEstacao();
}
```

```
public interface Relogio extends
AparelhoEletronico
{
    public void verHoras();
}
```



QUIZ: O código a seguir terá erro de compilação ou executará normalmente?

1.

```
public interface ICalculadora{
    public float somar(float v1, float v2){
        return v1 + v2;
    }

    public float subtrair(float v1, float v2){
        return  v1 - v2;
    }

    public float multiplicar(float v1, float v2){
        return  v1 * v2;
    }

    public float dividir(float v1, float v2){
        return  v1/v2;
    }
}
```

1. resposta

```
public interface ICalculadora{  
    public float somar(float v1, float v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1, float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float v2){  
        return v1/v2;  
    }  
}
```

Interfaces não podem conter
implementação de métodos!!

2.

```
public interface ICalculadora{  
    public void somar();  
    public void subtrair();  
    public void multiplicar();  
    public void dividir();  
}
```

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
}
```

2. resposta

```
public interface ICalculadora{  
    public void somar();  
    public void subtrair();  
    public void multiplicar();  
    public void dividir();  
}
```

A assinatura dos métodos deve ser igual
na interface e na classe que a
implementa!!

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
}
```

3.

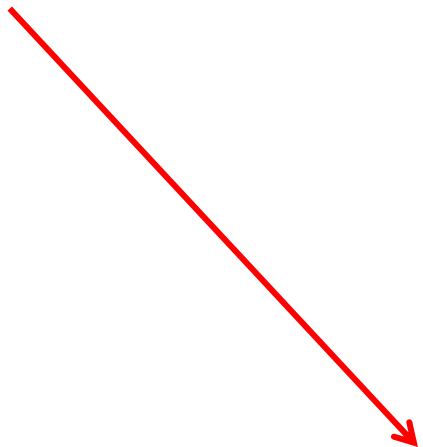
```
public interface ICalculadora{  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
    public float media(float v1,  
        float v2);  
}
```

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
}
```

3. resposta

```
public interface ICalculadora{  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
    public float media(float v1,  
        float v2);  
}
```

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
  
    ??  
}
```



4.

```
public interface ICalculadora{  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
}
```

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
  
    public float media(float v1, float  
        v2){  
        return (v1+v2)/2;  
    }  
}
```

4. resposta

```
public interface ICalculadora{  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
}
```



```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return  v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return  v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return  v1/v2;  
    }  
  
    public float media(float v1, float  
        v2){  
        return  (v1+v2)/2;  
    }  
}
```

5.

```
public interface ICalculadora{  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
    public float media(float v1,  
        float v2);  
}
```

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
  
    public float media(float v1, float  
        v2){  
        return (v1+v2)/2;  
    }  
}
```

5. resposta

```
public interface ICalculadora{  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
    public float media(float v1,  
        float v2);  
}
```



```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
  
    public float media(float v1, float  
        v2){  
        return (v1+v2)/2;  
    }  
}
```

6.

```
ICalculadora c = new ICalculadora();
```

6. resposta

```
ICalculadora c = new ICalculadora() ;
```

Não é possível instanciar um objeto
da Interface!!

7.

```
ICalculadora c = new Calculadora();
```

7. resposta

```
ICalculadora c = new Calculadora();
```



8.

```
public interface ICalculadora{  
    public ICalculadora();  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
}
```

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
  
    public float media(float v1, float  
        v2){  
        return (v1+v2)/2;  
    }  
}
```

8. resposta

```
public interface ICalculadora{  
    public ICalculadora();  
    public float somar(float v1,  
        float v2);  
    public float subtrair(float v1,  
        float v2);  
    public float multiplicar(float  
        v1, float v2);  
    public float dividir(float v1,  
        float v2);  
}
```

Interfaces não podem ter métodos construtores!!

```
public class Calculadora implements  
    ICalculadora{  
    public float somar(float v1, float  
        v2){  
        return v1 + v2;  
    }  
  
    public float subtrair(float v1, float  
        v2){  
        return v1 - v2;  
    }  
  
    public float multiplicar(float v1,  
        float v2){  
        return v1 * v2;  
    }  
  
    public float dividir(float v1, float  
        v2){  
        return v1/v2;  
    }  
  
    public float media(float v1, float  
        v2){  
        return (v1+v2)/2;  
    }  
}
```

9.

```
public interface ICalculadora{
    public float somar(float v1, float v2);
    public float subtrair(float v1, float
v2);
    public float multiplicar(float v1,
float v2);
    public float dividir(float v1, float
v2);
}
```

```
public class Aplicacao{

    public static void main(String
args[]){

        ICalculadora c = new
Calculadora();

        System.out.println(c.media(2,8)
);

    }

}
```

```
public class Calculadora implements
ICalculadora{
    public float somar(float v1, float
v2){
        return v1 + v2;
    }

    public float subtrair(float v1, float
v2){
        return  v1 - v2;
    }

    public float multiplicar(float v1,
float v2){
        return  v1 * v2;
    }

    public float dividir(float v1, float
v2){
        return  v1/v2;
    }

    public float media(float v1, float
v2){
        return  (v1+v2)/2;
    }

}
```

```
}
```

9. resposta

```
public interface ICalculadora{
    public float somar(float v1, float v2);
    public float subtrair(float v1, float v2);
    public float multiplicar(float v1, float v2);
    public float dividir(float v1, float v2);
    ??
}

public class Aplicacao{

    public static void main(String args[]){

        ICalculadora c = new Calculadora();

        System.out.println(c.media(2,8));
    }
}
```

```
public class Calculadora implements
    ICalculadora{
    public float somar(float v1, float v2){
        return v1 + v2;
    }

    public float subtrair(float v1, float v2){
        return v1 - v2;
    }

    public float multiplicar(float v1, float v2){
        return v1 * v2;
    }

    public float dividir(float v1, float v2){
        return v1/v2;
    }

    public float media(float v1, float v2){
        return (v1+v2)/2;
    }
}
```