

Programação Orientada a Objetos

Polimorfismo

Prof. Tito Kenzo



Polimorfismo

- Qualidade de possuir múltiplas formas, adj. polimorfo, multiforme;
- É responsabilidade do objeto que recebe a mensagem de executar a operação de modo adequado a sua classe;
- Esconde implementações de uma operação com uma interface comum;
- Significa que a mesma operação pode ter implementações diferentes.

Mais de um **Método** com o mesmo nome!

Polimorfismo

- Polimorfismo de Sobreposição (Overriding)
 - Uma classe pode sobrepor a implementação de uma operação que ela herda de uma superclasse.
 - **Mesma** assinatura do método da superclasse.
- Polimorfismo de Sobrecarga (Overloading)
 - Uma classe pode possuir mais de um método com o mesmo nome porém com assinatura diferente (número, tipo e ordem dos parâmetros).

Overriding

- Somente pode ser usado com a herança.
- Ocorre quando uma classe possui um método com o mesmo nome e assinatura (número, tipo e ordem dos parâmetros) de um método da superclasse.
- A sobreposição é resolvida dinamicamente (dynamic binding) em tempo de execução.

Overriding

Funcionario.java

```
public class Funcionario {  
    public void aplicaAumento(int percentual) { ... }  
}
```

Carteiro.java

```
public class Carteiro extends Funcionario{  
    public void aplicaAumento(int percentual){ ... }  
}
```

Overriding

Fruta.java

```
public class Fruta {  
    int gramas;  
    int cals_por_gramas;  
    void descascar()  
    {System.out.println("descascar uma fruta");}  
}
```

Citros.java

```
public class Citros extends Fruta{  
    void espremer() { /*.....*/}  
    void descascar()  
    {System.out.println("descascar uma citros");}  
}
```

Overloading

- Pode ser usado ou não com herança.
- Ocorre quando uma classe possui mais de um método com o **mesmo nome** porém com **assinatura diferentes** (número, tipo e ordem dos parâmetros).
- É resolvida dinamicamente (dynamic binding) pelo compilador em tempo de compilação. O compilador “escolherá” aquele que você realmente queria utilizar.

Overloading

Funcionario.java

```
class Funcionario {  
    public Funcionario (String nome) {...}  
    public Funcionario (String nome, int cpf) {...}  
}
```


Overloading

Fruta.java

```
public class Fruta {  
    int gramas;  
    int cals_por_gramas;  
    void descascar()  
    {System.out.println("descascar uma fruta");}  
}
```

Citros.java

```
class Citros extends Fruta{  
    void espremer() { /*.....*/}  
    void descascar(char quan)  
    {System.out.println("descascar "+quan+"citro(s)");}  
}
```

Tarefinha

- Criar uma classe Matematica com os seguintes métodos:
 - **min**: esse método deve poder receber de 2 ou 3 números e deve retornar o menor destes;
 - **max**: esse método deve poder receber de 2 ou 3 números e deve retornar o maior destes;

Obs.: “números” podem se referir a um tipo short, int, long, float ou double do Java.