



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

Laboratório Java

Implementação dos Conceitos Básicos da
Orientação a Objetos

Renata Mesquita (renatames@gmail.com)

Agenda

- Polimorfismo

Polimorfismo

- Em primeiro lugar, a palavra grega "Polimorfismo" significa várias formas em português.
- Em outras palavras, o conceito de polimorfismo indica as várias formas que vários objetos se comportam mesmo sendo todos eles oriundo de uma mesma classe, resumindo uma única classe pode dar origem a vários objetos cujos os comportamentos (métodos) podem ter implementações diferentes.

Polimorfismo

Qual a diferença entre o polimorfismo override e o overload no java

1º) Overload, palavra inglesa que em português significa sobrecarga. A sobrecarga de métodos consiste em voce declarar vários métodos em uma única classe ou em classes descendentes, com o mesmo nome, porém eles tem que ter assinaturas diferentes. A assinatura do método é o escopo do método identificado pelo seu retorno, pelo seu nome, bem como o recebimento ou não de seus parâmetros. Vamos ao exemplo:

```
public class Teste {  
    public void fazAlgo() {  
        System.out.println("Este método não recebe parâmetro");  
    }  
    public void fazAlgo(String mensagem) {  
        System.out.println("Mensagem");  
    }  
}
```

Repare que os métodos da classe Teste, tem o mesmo nome, porém eles serão identificados no momento de serem invocados, pela sua assinatura, ou seja a passagem de parâmetros ou não passagem de parâmetros.

Como disse antes, a sobrecarga de métodos voce pode fazer tanto na mesma classe, como também nas classes herdadas, bastando tão somente voce saber qual é a assinatura do método da super classe para poder sobrecarregá-lo.

Polimorfismo

Qual a diferença entre o polimorfismo override e o overload no java

2º) Override, palavra inglesa que significa "Sobrescrita" em português. A sobrescrita somente é possível em classes herdadas, isto porque este conceito significa o que o próprio nome já diz, ou seja, substituir um método da super classe na sub classe sobrescrevendo-o (substituindo-o) o mesmo. Vamos ao exemplo.

```
public class Teste {  
    public void fazAlgo() {  
        System.out.println("Este é o método da super classe");  
    }  
}  
  
public class NovoTeste extends Teste {  
    @Override  
    public void fazAlgo() {  
        System.out.println("Este é o método foi sobrescrito");  
    }  
}
```

Repare que na sub classe NovoTeste eu estou substituindo o método fazAlgo da super classe, isto porque, se eu não o sobrescrevesse, ele iria herdar o método fazAlgo() da super classe, e neste caso ele passa a ter o método com o mesmo nome e mesma assinatura só que com implementação nova na sub classe.

A sintaxe "`@Override`" é uma diretiva de compilação, que orienta o compilador que as linhas abaixo de sua declaração contém um método igual da super classe que deverá ser substituído.

Como disse antes, a sobrescrita só é permitida em classes herdadas, isto pois, métodos com a mesma assinatura e nomes iguais não podem ter na mesma classe.

Polimorfismo

Qual a diferença entre o polimorfismo override e o overload no java

Valendo lembrar que se voce instanciar um objeto do tipo da super ele enxerga apenas o seu método faz algo, vamos ao exemplo.

```
public class Teste {  
    public void fazAlgo() {  
        System.out.println("Este é o método da super classe");  
    }  
}  
  
public class NovoTeste extends Teste {  
    @Override  
    public void fazAlgo() {  
        System.out.println("Este é o método foi sobrescrito");  
    }  
}  
  
public static void main(String args[]){  
    Teste T = new Teste();  
    T.fazAlgo(); // Retorna "Este é o método da super classe"  
    TesteNovo = TN = new TesteNovo();  
    TN.fazAlgo(); // Retorna "Este método foi sobrescrito"  
}
```

Polimorfismo

Qual a diferença entre o polimorfismo override e o overload no java

O porque que não existe a diretiva `@Overload` (Sobrecarga) no Java :?: :?: :?:

Simple, o compilador javac consegue identificar métodos cujas as assinaturas são diferentes, porém não conseguem identificar os com assinaturas iguais, daí a necessidade da sintaxe "`@Override`".

Bibliografia

- JAVA e Orientação a Objetos – Caelum Ensino e Soluções em JAVA