

Polimorfismo - Davi e Marcelo

Definições

Pode ser definida como o princípio a partir do qual as classes derivadas de uma classe, são capazes de invocar os métodos; Os mesmos atributos e objetos podem ser utilizados em objetos distintos, com implementação e lógicas diferentes.

Referencias:

<https://www.javaworld.com/article/3033445/learn-java/java-101-polymorphism-in-java.html>

<http://www.devmedia.com.br/conceitos-e-exemplos-polimorfismo-programacao-orientada-a-objetos/18701>

Exemplos

Exemplo 1 -

```
public class Animal{
    public void sound(){
        System.out.println("Animal is making a sound");
    }
}

class Horse extends Animal{
    @Override
    public void sound(){
        System.out.println("Neigh");
    }
    public static void main(String args[]){
        Animal obj = new Horse();
        obj.sound();
    }
}
```

Resultado: **Neigh**

(<https://beginnersbook.com/2013/03/polymorphism-in-java/>)

Exemplo 2 -

```
public class Animal {
    public void comer() {
        System.out.println( "Animal Comendo..." );
    }
}

public class Cao extends Animal {
    public void comer() {
        System.out.println( "Cão Comendo..." );
    }
}

public class Tigre extends Animal {
    public void comer() {
        System.out.println( "Tigre Comendo..." );
    }
}

public class Test {
    public void fazerAnimalComer( Animal animal ) {
        animal.comer();
    }

    public static void main( String[] args ) {
        Test t = new Test();
        t.fazerAnimalComer( new Animal() );
        t.fazerAnimalComer( new Cao() );
        t.fazerAnimalComer( new Tigre() );
    }
}
```

Resultado : Animal Comendo...
Cão Comendo...
Tigre Comendo...

(<http://www.guj.com.br/t/exemplos-praticos-de-polimorfismo/65598/2>)

Ferramenta

1. Caso a classe pai ter um método abstrato, verificar se na classe filha o mesmo foi implementado.
2. Verificar se os comportamentos dos métodos nas classes filhas são distintos.
3. A utilização do cast down não é recomendada;

Inspeção

Classes devem ser inspecionadas com a ferramenta
Referenciar origem

Exemplo 1 - Java.util.Calendar()

- 1- Falso
- 2 - Falso
- 3 - Verdadeiro

Exemplo 2 - Java.util.Collection()

- 1 - Falso
- 2 - Falso
- 4 - Verdadeiro

Exemplo 3 -

(<https://github.com/zoqfabio/Empregos/blob/master/src/java/br/edu/unidavi/empreabilidade/util/servlets/usuario/UsuarioSalvar.java>)

- 1 - Verdadeiro
- 2 - Verdadeiro
- 3 - Verdadeiro

Exemplo - 4

(<https://github.com/eldermoraes/javaoneus2016/blob/master/app/javaonedb/src/main/java/com/javaone/servlet/DataServlet.java>)

- 1 - Verdadeiro
- 2 - Verdadeiro

3 - Verdadeiro

Exemplo - 5

(<https://github.com/Wrakor/GarbageCollector/blob/master/code/GarbageCollector/src/main/ContainerThread.java>)

1 - Verdadeiro

2 - Verdadeiro

3 - Verdadeiro