

Curso Profissional: Programação de Informática

PSD – 11.º ano: UFCD 0816 - Programação de sistemas distribuídos - JAVA

Ficha de Trabalho 11

Ano letivo 22/23

Operador instanceof

Uma aplicação em Java facilmente atinge dezenas ou centenas de classes, subclasses, superclasses, classes abstratas e outros nomes da Orientação a Objetos.

Através do nome do objeto instanciado pode ser complicado ver a que classe ele pertence. A resolução do problema faz-se usando o operador instanceof.

O operador instanceof em java é usado para testar se o objeto é uma instância do tipo especificado (classe ou subclasse ou interface) ->ver ficha 2 no que se refere a operadores relacionais.

O instanceof em java é também conhecido como operador de comparação do tipo porque ele compara a instância de tipo. Ela retorna true ou false. Ao aplicar o operador instanceof, com qualquer variável que tem valor nulo, ele retorna false.

Síntaxe:

nomeObjeto instanceof Classe

Exemplo 1:

```
class Simple1{
    public static void main(String args[]){
        Simple1 s=new Simple1();
        System.out.println(s instanceof Simple1); //true
    }
}
```

Exemplo 2:

```
class Animal{
}
class Dog1 extends Animal{//Dog inherits Animal

    public static void main(String args[]){
        Dog1 d=new Dog1();
        System.out.println(d instanceof Animal); //true
    }
}
```

Exemplo 3:

-->zoo.java

```
public class zoo {

    public static void main(String[] args) {
        Vaca vaca = new Vaca();
        Gato gato = new Gato();
        Carneiro carneiro = new Carneiro();

        Animal bichos[] = {vaca, gato, carneiro};

        for(int i=0 ; i < bichos.length ; i++)
        {
            if(bichos[i] instanceof Vaca){
                System.out.print("A vaca tem " +
bichos[i].numeroPatas + " patas e faz ");
                bichos[i].som();
                System.out.println();
            }

            if(bichos[i] instanceof Gato){
                System.out.print("O gato tem " +
bichos[i].numeroPatas + " patas e faz ");
                bichos[i].som();
                System.out.println();
            }

            if(bichos[i] instanceof Carneiro){
                System.out.print("O carneiro tem " +
bichos[i].numeroPatas + " patas e faz ");
                bichos[i].som();
                System.out.println();
            }
        }
    }
}
```

-->Animal.java

```
public class Animal {
    protected String nome;
    protected int numeroPatas;

    public void som() {
    }
}
```

-->Vaca.java

```
public class Vaca extends Animal {
    public Vaca() {
        this.nome = "Mimosa";
        this.numeroPatas = 4;
    }

    public void som() {
        System.out.print("MUUUU");
    }
}
```

-->Gato.java

```
public class Gato extends Animal{
    public Gato() {
        this.nome = "Bichano";
        this.numeroPatas = 4;
    }

    public void som() {
        System.out.print("MIAU");
    }
}
```

-->Carneiro.java

```
public class Carneiro extends Animal{
    public Carneiro() {
        this.nome = "Barnabé";
        this.numeroPatas = 4;
    }

    public void som() {
        System.out.print("BÉÉÉ");
    }
}
```

Bibliografia

<http://www.javatpoint.com/downcasting-with-instanceof-operator>

<http://www.javaprogressivo.net/2012/10/descobrimos-classe-de-um-objeto.html>

EXERCÍCIOS

1. Escreve o código referente ao exemplo 3.

2. Usa o operador instanceof para rescrever o método ListarDetalhes(), da classe TestarPersonagens criada na ficha de trabalho n.º 10. Dá-lhe o nome de ListarDetalhes2.

```
if (p instanceof Vampiro){  
    ((Vampiro) p).Suga();  
}
```

Faz a chamada do método no } programa principal e executa de novo.