

Comparando objetos

DigitalHouse>



**Certified Tech
Developer**

The Ultimate Degree

Índice

1. Comparando objetos
2. Interface Comparable
3. Implementação em Java

1 | Comparando objetos



“



Hora de aprendermos boas práticas de
como comparar objetos!

”

Interfaces

Ao comparar tipos primitivos, fazemos isso com os operadores "==", ">", "<", ">=", "<=", "!", "! =", mas como podemos fazer se quisermos comparar dois objetos? Por exemplo, dois pimentões?

```
int i = 5;  
int j = 6;  
If(i < j)  
    System.out.println("i <  
j");  
else  
    System.out.println("j  
>= i");
```



Interfaces

Para comparar dois **objetos**, a primeira coisa que devemos saber é por **qual** ou **quais** de seus atributos iremos compará-los. Ou seja, como respondemos à pergunta: **esses pimentões são iguais?**



A primeira questão que vai surgir é se devemos considerar a cor, tipo, peso ou tamanho e substituir o método `equals()`.

Interfaces



O método `equals()` só nos serve para comparar a igualdade, mas não conseguimos dizer se é maior ou menor que outro objeto.

Método compareTo

Uma solução para o problema levantado é **garantir** que todos os objetos que precisam comparar tenham, por exemplo, um **método compareTo** que recebe como **parâmetro** o outro objeto com o qual se deseja fazer a comparação e retorna, por exemplo:

- **Zero:** se forem iguais.
- **Maior que zero:** se o objeto que invoca o método for maior que o recebido como parâmetro.
- **Menor que zero:** se o objeto que invoca o método for menor que o recebido como parâmetro.

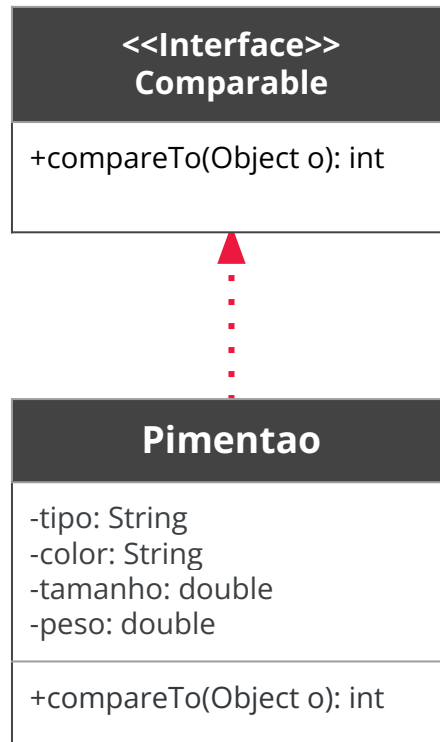


Método compareTo

Como forçamos todos os objetos que queremos comparar a ter um método **compareTo**?



Com as interfaces, podemos fazer com que: quem as implementa tenha um método `compareTo` e possa estabelecer sua própria implementação.



2 | Interface Comparable

Interface Comparable

Não precisamos criar uma interface para comparar objetos porque **Java** tem a sua própria, é a **interface Comparable** e é necessário usá-la em outras circunstâncias para comparar objetos, por exemplo, para ordená-los em **coleções**.

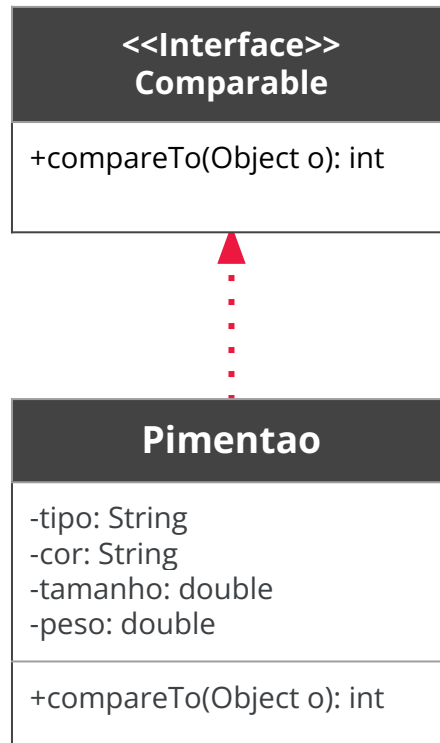


Interface Comparable

O método que reforça a interface Java Comparable é o método **compareTo**.



Para usar a interface Comparable do Java, devemos importar o pacote `java.lang`.



3 | Implementação em Java

Implementação em Java

```
import java.lang.*;
public class Pimentao implements Comparable{

    private String tipo;
    private String color;
    private double tamanho;
    private double peso;

    public Pimentao(){}
    public int compareTo(Object obj){

        Pimentao p2 = (Pimentao) obj;
        int resposta = 0;

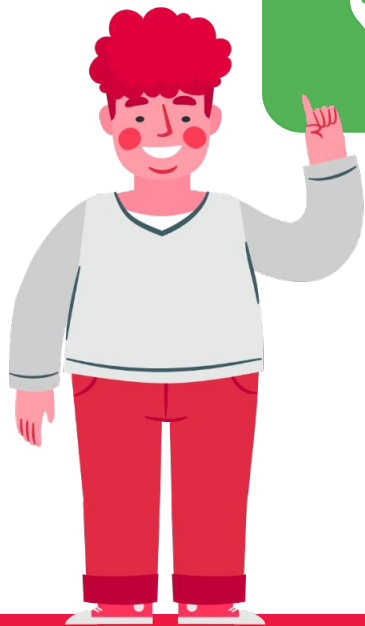
        if(this.getPeso() > p2.getPeso())
            resposta = 1;

        if(this.getPeso() < p2.getPeso())
            resposta = -1;

        return resposta;

    }
}
```

Implementação em Java



O método `compareTo` deve retornar:

- Se eles forem iguais: 0.
- Se for maior: um número maior que zero.
- Se for menor: um número menor que zero.

Implementação em Java

```
public class Teste {  
    public void main(String args[]){  
  
        Pimienta p1 = new Pimentao();  
        p1.setPeso(200);  
        p1.setCor("amarelo");  
        Pimienta p2 = new Pimentao();  
        p2.setCor("vermelho");  
        p2.setPeso(150);  
  
        if(p1.compareTo(p2) > 0){  
            System.out.println("O pimentao amarelo é maior que o vermelho");  
        }else if(p1.compareTo(p2) < 0){  
            System.out.println("O pimentao vermelho é maior que o amarelo");  
        }else{  
            System.out.println("O pimentao vermelho é igual ao amarelo");  
        }  
    }  
}
```


Implementação em Java



Finalmente, pela nossa implementação, podemos dizer que o pimentão amarelo é maior que o vermelho.

DigitalHouse>
Coding School