



Aula 10

Interfaces e Comparadores

Iniciativa Conjunta:



Com o apoio de:





Interfaces

- Definem um comportamento que as classes podem implementar.
- Contêm apenas constantes e declarações de métodos.
- Uma interface não é uma classe.

```
public interface NomeInterface1 {  
  
    ...  
  
}
```

- Uma classe pode implementar várias interfaces.
- Uma interface pode estender várias interfaces.
- Se uma classe implementar uma interface é necessário que todos os métodos da interface sejam definidos na classe. Caso contrário o compilador irá dar erro.



Interfaces

- Esta relação é especificada no cabeçalho da classe que assume implementar uma determinada interface através da palavra reservada `implements`.

```
public class NomeClass implements NomeInterface1 {  
    ...  
}
```

- Caso a classe implemente mais do que uma interface basta incluir os respectivos nomes separados por vírgulas.

```
public class NomeClass implements NomeInterface1, NomeInterface2 {  
    ...  
}
```



Exemplo 1

```
public interface FiguraGeometricaPlana {  
    public String getNomeFiguraPlana();  
    public int getArea();  
    public int getPerimetro();  
}  
  
public class Quadrado implements FiguraGeometricaPlana{  
    private int lado;  
    public int getLado(){  
        return lado;  
    }  
    public void setLado(){  
        this.lado=lado;  
    }  
  
    ...  
}
```



Exemplo 1 - cont.

...

```
@Override
public int getArea(){
    return lado * lado;
}

@Override
public int getPerimetro(){
    int perimetro = lado * 4;
    return perimetro;
}

@Override
public String getNomeFiguraPlana(){
    return "Quadrado";
}
}
```



Interfaces vs. Classes Abstratas

- Métodos da interface são **implicitamente abstratas**. Uma classe abstrata no entanto pode ter métodos implementados por defeito.
- Variáveis nas interfaces são por defeito **finais**.
- Membros das interfaces são **públicas** por defeito. Uma classe abstrata pode ser privada, protected, etc.
- A interface deve ser implementada com a chave “**implements**”, em vez do “**extends**”
- Interfaces só podem estender outras interfaces
- As classes (abstratas inclusive) podem estender uma única classe, mas implementar várias interfaces



Exercício 1

1. Criar as classes Rectângulo e Círculo que implementam a interface FiguraGeometricaPlana, como foi feito para a classe Quadrado.
2. Implementar a interface FiguraGeometricaTridimensional em todas as classes. Para o cálculo do volume e o nome todas as figuras tridimensionais irão ser prismas e um cilindro.

```
public interface FiguraGeometricaTridimensional {  
    public String getNomeFiguraTridimensinal();  
    public int getAltura();  
    public int getVolume();  
}
```



Comparable

- A interface Comparable é usada para ordenar objectos de uma classe definida pelo utilizador.
- Esta interface tem apenas um método denominado `compareTo(Object)`.
- O método é usado para comparar o objecto actual com o objecto especificado.
- Retorna: Inteiro positivo, se o objecto actual for maior que o objecto especificado. Inteiro negativo, se o objecto actual for menor que o objecto especificado. Zero, se o objecto for igual ao objecto especificado.



Exemplo 2

```
public Estudante implements Comparable<Estudante>{
    int numero;
    String nome;
    int idade;
    public Estudante(int numero, String nome, int idade){
        this.numero = numero;
        this.nome = nome;
        this.idade = idade;
    }
    public int compareTo(Estudante st){
        if(idade==st.idade)
            return 0;
        else if (idade>st.idade)
            return 1;
        else
            return -1;
    }
}
```



Exemplo 2 - Cont.

```
public static void main(String [] args){  
    Estudante [] st=new Estudante[3];  
    st[0]=new Estudante(101,"José",23);  
    st[1]=new Estudante(106,"Maria",27);  
    st[2]=new Estudante(105,"Manuel",21);  
  
    Arrays.sort(st);  
  
    System.out.println("Ordenado por idade");  
  
    for(Estudante stu:st){  
        System.out.println(stu.numero + " " + stu.nome + " " +  
            stu.idade);  
    }  
}
```

Resultado na consola:

Ordenado por idade

105 Manuel 21

101 José 23

106 Maria 27



Exercício 2

Criar a classe Filme que vai ter os seguintes atributos: pontuação, nome e ano. A classe Filme vai implementar a interface Comparable:

1. Ordene os seguintes filmes por ordem crescente do seu ano e depois
2. Ordene os filmes por ordem decrescente em relação à sua pontuação

Filmes:

- Force of Awakens, 8, 2015
- Star Wars, 9, 1977
- Empire Strikes Back, 7, 1980
- Return of the Jedi, 10, 1983



Comparator

- A interface Comparator é usada para ordenar objectos de uma classe definida pelo utilizador.
- Esta interface tem o método `compare(Object obj1, Object obj2)`.
- Ao contrário do Comparable, o Comparator é externo ao tipo de objecto que está a ser comparado.
- Podem ser criadas classes separadas (que implementam o Comparator) para comparar por atributos diferentes.



Exemplo 3

```
public class NumeroCompare implements Comparator<Estudante> {  
    public int compare(Estudante s1, Estudante s2){  
        if (s1.numero < s2.numero) {  
            return -1;  
        } else if (s1.numero > s2.numero) {  
            return 1;  
        } else {  
            return 0;  
        }  
    }  
}
```

```
public class NomeCompare implements Comparator<Estudante> {  
    public int compare(Estudante s1, Estudante s2){  
        return s1.nome.compareTo(s2.nome);  
    }  
}
```



Exemplo 3

Continuação com o main do Exemplo 2:

```
NomeCompare nomeCompare = new NomeCompare();
Arrays.sort(st, nomeCompare);
System.out.println("Ordenado por nome");
for(Estudante stu:st){
    System.out.println(stu.numero + " " +
        stu.nome + " " + stu.idade);
}

NumeroCompare numCompare=new NumeroCompare();
Arrays.sort(st, numCompare);
System.out.println("Ordenado por numero");
for(Estudante stu:st){
    System.out.println(stu.numero + " " +
        stu.nome + " " + stu.idade);
}
```

Resultado na consola:

Ordenado por idade

105 Manuel 21

101 José 23

106 Maria 27

Ordenado por nome

106 Maria 27

105 Manuel 21

101 José 23

Ordenado por numero

101 José 23

105 Manuel 21

106 Maria 27



Exercício 3

Criar a classe Livro que tem os seguintes atributos: Nome, Ano, Editora e Autor. Implementar as interfaces relativas a comparadores de modo a ordenar o nome e o autor por ordem alfabética, a editora por ordem contrária à ordem alfabética e o ano por ordem decrescente.

Livros:

- The Shinning, 1990, Leya, Stephen King
- Harry Potter e a Pedra Filosofal, 2001, ASA, Joanne Rowling
- O Inferno de Dante, 1472, Bertrand, Dante
- Conde de Monte Cristo, 1844, Texto, Alexandre Dumas
- Desenhar Bases de Dados, 2016, Silabo, Pedro Nogueira