



**TECNOLOGIA E
INOVAÇÃO EM PROL DA
INDÚSTRIA**



Curso Técnico em Informática

Desenvolvimento de Sistemas I - 100h

Prof^a: Francisleide Almeida

Classe abstrata

- Se não faz sentido instanciar, criaremos uma classe abstrata.
- No exemplo anterior, faria sentido criar um objeto do tipo animal?

Animal a1 = new Animal();

- Não!
- Animais, serão sempre de algum tipo: cachorro, vaca, gato, etc..
- Então podemos considerar a classe Anima, uma classe abstrata.

Métodos abstratos

- Métodos abstratos só serão criados quando você tiver certeza que eles não serão chamados pelos objetos.
- Qualquer classe que estender a classe abstrata será obrigada a reescrever este método;

Exemplo

```
public abstract class Conta { private double
    saldo;

    public void setSaldo(double saldo) {
        this.saldo = saldo;
    }
    public double getSaldo() {
        return saldo;
    }
    public abstract void imprimeExtrato();

}
```

```
Public ContaPoupanca extends Conta { @Override
    public void imprimeExtrato() {
        System.out.println("### Extrato da Conta ###");
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/aaaa HH:mm:ss");
        Date date = new Date();
        System.out.println("Saldo: "+this.getSaldo()); System.out.println("Data:
        "+sdf.format(date));
    }
}
```


Interface

- A interface é um recurso muito utilizado para “obrigar” a um determinado grupo de classes a ter métodos ou propriedades em comum para existir em um determinado contexto;
- É como um contrato que quando assumido por uma classe deve ser implementado.

Interface

- Apenas assinaturas de métodos e propriedades;
- A classe que a utilizará realiza a implementação das assinaturas, dando comportamentos práticos aos métodos.

Interface

- Ao contrário da herança que limita uma classe a herdar somente uma classe pai por vez, é possível que uma classe implemente varias interfaces ao mesmo tempo.

Exemplo

- Vamos supor que cientistas te contratem para criar um aplicativo Java para uma universidade, para fazer cálculos com matrizes. Eles irão te passar o que você tem que fazer, porém, muito provavelmente não vão estar interessados em COMO você vai fazer isso, só querem saber se vai funcionar.

```
public interface Matriz {  
    double Traco(Matriz m);  
    Matriz Nula(int linha, int coluna);  
    Matriz Transposta(Matriz m);  
}
```

```

public class minhaMatriz implements Matriz{  Matriz mat;
double traco;

public double Traco(Matriz m){
    //Implementação do traço de uma matriz
    return traco;
}

public Matriz Nula(int linha, int coluna){
    //Zerando todos os elementos da matriz
    return mat;
}

public Matriz Transposta(Matriz m){
    //Armazenando a transposta da matriz 'm' na matriz 'mat'
    return mat;
}
}
    
```


Exemplo 2

```
public interface Conta{  
    void depositar(double valor);  
    void sacar(double valor);  
    double getSaldo();  
  
}
```

Exemplo 2

```
public class ContaCorrente implements Conta { private
    double saldo;
    private double taxaOperacao = 0.45;
    @Override
    public void deposita(double valor) {
        this.saldo += valor - taxaOperacao;
    }
    @Override
    public double getSaldo() {
        return this.saldo;
    }
    @Override
    public void sacar(double valor) {
        this.saldo -= valor + taxaOperacao;
    }
}
```



Exemplo 2

```
public class ContaPoupanca implements Conta {
    private double saldo;

    @Override
    public void deposita(double valor) {
        this.saldo += valor;
    }

    @Override
    public double getSaldo() {
        return this.saldo;
    }

    @Override
    public void sacar(double valor) {
        this.saldo -= valor;
    }
}
```

Regras:

- Os métodos abstratos definidos em uma classe abstrata devem obrigatoriamente ser implementados em uma classe concreta.
- Mas se uma classe abstrata herdar outra classe abstrata, a classe que herda não precisa implementar os métodos abstratos.

Exemplos

```

01. package classeAbstrata;
02.
03. public abstract class Eletrodomestico {
04.     private boolean ligado;
05.     private int voltagem;
06.
07.     // métodos abstratos //
08.     /*
09.      * não possuem corpo, da mesma forma que
10.      * as assinaturas de método de uma interface
11.      */
12.     public abstract void ligar();
13.     public abstract void desligar();
14.
15.     // método construtor //
16.     /*
17.      * Classes Abstratas também podem ter métodos construtores,
18.      * porém, não podem ser usados para instanciar um objeto diretamente
19.      */
20.     public Eletrodomestico(boolean ligado, int voltagem) {
21.         this.ligado = ligado;
22.         this.voltagem = voltagem;
23.     }
24.
25.     // métodos concretos
26.     /*
27.      * Uma classe abstrata pode possuir métodos não abstratos
28.      */
29.     public void setVoltagem(int voltagem) {
30.         this.voltagem = voltagem;
31.     }
32.
33.     public int getVoltagem() {
34.         return this.voltagem;
35.     }
36.
37.     public void setLigado(boolean ligado) {
38.         this.ligado = ligado;
39.     }
40.
41.     public boolean isLigado() {
42.         return ligado;
43.     }
44. }

```

```

01. package classeAbstrata;
02.
03. public class TV extends Eletrodomestico {
04.     private int tamanho;
05.     private int canal;
06.     private int volume;
07.
08.     public TV(int tamanho, int voltagem) {
09.         super (false, voltagem); // construtor classe abstrata
10.         this.tamanho = tamanho;
11.         this.canal = 0;
12.         this.volume = 0;
13.     }
14.
15.     /* implementação dos métodos abstratos */
16.     public void desligar() {
17.         super.setLigado(false);
18.         setCanal(0);
19.         setVolume(0);
20.     }
21.
22.     public void ligar() {
23.         super.setLigado(true);
24.         setCanal(3);
25.         setVolume(25);
26.     }
27.
28.     // abaixo teríamos todos os métodos construtores get e set...
29.
30. }

```


Atividade

- Agora crie um outro eletrodoméstico chamado “Rádio”, onde existam os atributos fixos: AM = 1 e FM = 2, banda (am ou fm), sintonia (float) e volume (int). Crie um construtor para esta classe, que além de passar os valores da classe do construtor da super classe, inicialize a banda como FM e a sintonia e o volume como 0.

Continuação

- Crie uma classe principal para instanciar um Rádio e uma TV.
- Posso instanciar a classe Eletrodoméstico? Porque?



Você foi convidado a fazer um sistema de uma empresa artística que trabalha em um teatro. Neste teatro tem-se apresentações de artistas que serão: dançarinos, cantores ou atores.

Todos eles possuem: nome, cpf, ano de nascimento e tempo de serviço.

Os atores tem um diferencial. Para o teatro é importante a informação se ele é formado ou não, para obter mais benefícios.

Um cantor deve possuir uma banda e a banda contém um guitarrista, o um baterista e um baixista. Todos eles com nome, cpf, ano de nascimento e tempo de serviço.

Este teatro utiliza um esquema de pontuação para todos os artistas, algumas ações que eles fazem acumula pontos e no final estes pontos são convertidos em benefícios, como cortesias para espetáculos, shows e prêmios, mas de acordo com sua especialidade, a forma de acúmulo de pontos é diferenciada.

Para o dançarino, o sistema perguntará quantos espetáculos ele já realizou naquele teatro, se ele tiver participado de mais de 5 apresentações, ele ganha 100 pontos;

Já o cantor, se ele tiver mais de 18 anos (calcule a idade dele de acordo com o ano atual de 2018) e se apresentar mais de 3 vezes no teatro, ele acumula 100 pontos.

Por último, para o ator acumular os 100 pontos, ele precisa estar formado e fazer 3 apresentações no teatro.

O teatro possui um esquema de que, quando atinge 1000 pontos, somados dos pontos de todos os cantores, dançarinos, etc. são fornecidos um show gratuito.