

Lista de exercícios sobre interface

1) Uma TV é caracterizada com os atributos tamanho, canal e volume – todos do tipo *int* – e o atributo ligada (*boolean*). Seu construtor recebe apenas o valor do tamanho e o resto dos atributos inteiros é iniciado com valor ZERO. O atributo ligada se inicia com false.

Existem dois modelos de TV, subclasses de TV, que também serão caracterizadas como classes: TV001 e SDX. A TV001 é uma TV simples, sem muitos recursos que quando acionarmos o comando desligar irá simplesmente desligar. Já o modelo SDX é uma TV mais moderna, que quando acionarmos o comando desligar irá apresentar uma mensagem dizendo "Tchau!".

Como geralmente toda ação da TV é executada pelo controle remoto, então iremos criar uma interface de controle remoto, que deve ser implementada por ambos os modelos.

```
public interface ControleRemoto {  
    void mudarCanal(int canal);  
    void aumentarVolume();  
    void diminuirVolume();  
    void ligar();  
    void desligar();  
}
```

As funções de aumentar e diminuir volume devem respectivamente incrementar e decrementar o volume em uma unidade de TV.

Em outra classe independente, elabore um método *main* que instancie as duas TVs, ligue-as, aumente e diminua volume, mude de canal e depois desligue ambas.

IMPORTANTE: Todos os atributos das classes descritas devem ser criados como *private* e todos eles devem possuir métodos acessores (*get* e *set*).

2) A classe Mapa encapsula os dados de um mapa (como tipos de documentos em bibliotecas). Para simplificar, somente alguns atributos devem ser definidos nesta classe: titulo (*String*), escala (*double*) e descricaoSumaria (*String*). Seu único construtor deve receber sempre estes 3 atributos como parâmetros. Possui também um método *toString()*, que reúne todas as informações do objeto da classe.

A classe MapaDeBiblioteca é uma subclasse de Mapa, contendo adicionalmente dois atributos: emprestado (*boolean*) e localizacao (*String*), sendo este último possuindo um valor constante de “Mapoteca”. O seu construtor recebe os mesmos parâmetros do construtor da sua superclasse. Esta classe deve implementar a interface ItemDeBiblioteca, descrita abaixo. O método *descricao()* deve retornar as informações do objeto desta classe. Utilize o método *toString()* da classe pai para auxiliar este método.

```
public interface ItemDeBiblioteca {
    public final static int máximoDeDiasParaEmpréstimo = 14;
    public void empresta();
    public void devolve();
    String descricao();
}
```

Existe uma classe Livro que contém informações gerais de um livro. Possui os atributos título (String), autor (String), numeroDePaginas (short) e anoDaEdicao (short). Seu único construtor deve receber sempre estes 4 atributos como parâmetros. Possui também um método *toString()*, que reúne todas as informações do objeto da classe.

A classe LivroDeBiblioteca é uma subclasse de Livro, contendo adicionalmente dois atributos: *emprestado* (*boolean*) e *localizacao* (*String*). O seu construtor recebe os mesmos parâmetros do construtor da sua superclasse. Esta classe deve implementar a interface ItemDeBiblioteca, definida anteriormente. O método *descricao()* deve retornar as informações do objeto desta classe. Utilize o método *toString()* da classe pai para auxiliar este método.

IMPORTANTE: Todos os atributos das classes descritas devem ser criados como *private* e todos eles devem possuir métodos acessores (*get* e *set*).