

Exercícios Interface e Herança

Fátima L. S. Nunes



Exercício1

Uma administradora de condomínios deseja obter informações sobre os condomínios que administra e seus moradores. Para isso é definida uma classe Apartamento, da seguinte forma:

```
Classe Apartamento
{
    int qtdeMoradores; // armazena a quantidade de
    moradores no apartamento
    String nomeResponsavel; // nome da pessoa
    responsável pelo apartamento
}
```



Exercício 1

Suponha que esta classe já forneça os seguintes métodos:

Apartamento(int qm, String n) - construtor

int obtemQM () - devolve a quantidade de moradores

String obtemNR() - devolve o nome do responsável

void imprimeDados() - imprime os atributos da classe

Exercício 1

Faça duas classes denominadas, respectivamente, AptoComum e Cobertura, que são filhas da classe Apartamento.

A classe AptoComum tem os atributos adicionais qteDormitórios (int), temEscritório (boolean) e vagasGaragem (int).

A classe Cobertura tem os atributos adicionais tamanhoChurrasqueira (double) e temPiscina (boolean).

Essas classes devem ter um método imprimeDados() que redefine o método da classe mãe e imprime também os dados adicionais inerentes a cada uma delas.

As duas classes devem ter construtores que recebem os dados particulares de cada uma delas.

Exercício 2

Acrescente à classe Apartamento os atributos valor (double), númeroMaximoPrestações (double) e formaDePagamento (String). Esses atributos não podem ter acesso direto e você deve implementar métodos de acesso público a eles.

Acrescente à classe Apartamentos os seguintes métodos, com as seguintes especificações:

- **defineValor** - método que não pode ser sobrecarregado nas classes filhas. Recebe um valor com parâmetro e atualiza o atributo correspondente;
- **definePagamento** - método que será implementado somente nas classes filhas. Na classe AptoComum, deve receber um parâmetro com o número de prestações, atualizar o atributo correspondente e atualizar a “formaDePagamento” para “parcelado”. Na classe Cobertura, o número de prestações é igual a zero e a forma de pagamento é “à vista”.

Exercício 3

Acrescente à classe Apartamento dois atributos do tipo String: vantagem e desvantagem. Esses atributos não podem ter acesso direto e você deve implementar métodos de acesso público a eles. Os métodos somente de acesso poderão ser vistos somente pelas classes que estão no mesmo pacote que a classe Apartamento ou por suas filhas.

Exercício 4

Considere as classes Livro e Biblioteca, dadas a seguir. Deseja-se imprimir duas listagens distintas, sendo a primeira em ordem alfabética crescente de título e a segunda em ordem crescente de ano. Utilizando a interface ComparadorDeLivro dada a seguir, faça:

- Duas classes que implementem a interface fornecida e que sejam usadas para imprimir as listagens solicitadas.
- Um método para ordenar um vetor de Livros . Este método deve utilizar obrigatoriamente objetos das classes criadas no item anterior, com conceito de interface, para ordenar o vetor de acordo com o atributo solicitado (título ou ano).
- Complete o método **main** da classe Biblioteca nos locais indicados para imprimir as listagens solicitadas.



Exercício 4

```
class Livro
{
    private String titulo;
    private int ano;
    private int tipo; // valor de 1 a 20 que classifica a categoria de
cada livro

    void setTitulo (String t) {titulo= t;}
    void setAno (int a) {ano = a;}
    void setTipo (int ti) {tipo = ti;}
    String getTitulo () {return titulo;}
    int getAno () { return ano;}
    int getTipo () { return tipo;}
}
```

```
interface ComparadorDeLivro
{
    int livroMenor(Livro livro1, Livro livro2)
    // retorna: 1 se atributo específico de livro1 é maior que livro2
    //          -1 se atributo específico de livro1 é menor que livro2
    //          0 se atributo específico de livro1 é igual a livro2
}
```



Exercício 4

```
class Biblioteca
{
    // método para criar vetor de Livros
    // aqui vai o código para criar o vetor de Livros (você não precisa
completar)

    // método imprime
    // aqui vai o código para imprimir o vetor de Livros (você não precisa
completar)

    // método para ordenar o vetor
    ... aqui vai código para ordenar o vetor (você deve apresentá-lo completo-)

    // método principal
public static void main (String [] args)
{
    Biblioteca bibl= new Biblioteca();

    System.out.println("Livros desordenados");
    bibl.imprime();

    System.out.println("Livros em ordem de título")
    (você deve completar aqui)

    System.out.println("Livros em ordem de ano")
    (você deve completar aqui)
}
```



Exercícios Interface e Herança

Fátima L. S. Nunes



EACH