

## 2ª Lista de Exercícios

1. Crie uma classe `Animal` para um Pet shop. Essa classe deverá utilizar encapsulamento, possuir informações sobre a espécie do animal (cachorro, gato, etc), sua idade, se está vacinado ou não, seu preço e um campo descrição para outras informações (ex: raça, problema físico etc). Defina também um construtor.

Crie uma classe `Programa` com o método `main` e construa objetos da classe `animal`, imprimindo os valores de seus atributos na tela.

2. Crie uma classe que represente um retângulo. Esta classe deve utilizar encapsulamento, conter métodos para calcular área e perímetro bem como um método `toString()` que retorna uma string com todas as informações do retângulo (valor das variáveis internas, área e perímetro).

Defina uma classe `Programa` com o método `main` e crie objetos `retângulo`, imprimindo suas informações na tela.

3. Defina uma classe `Java` para representar uma pessoa. Uma pessoa, no nosso caso, deverá possuir os seguintes atributos: Nome, Idade, Altura, Peso e Sexo.

A classe deve usar encapsulamento e possuir um método para calcular o IMC (Índice de Massa Corpórea) e um método `toString()` que fornece o nome da pessoa e uma descrição de sua categoria de acordo com o IMC. O IMC é obtido através da divisão do peso pela altura ao quadrado e é interpretado da seguinte forma:

IMC  $\leq$  18,5: Abaixo do peso normal  
18,5 < IMC  $\leq$  25: Peso Normal  
25 < IMC  $\leq$  30: Acima do peso normal.  
IMC > 30: Obesidade.

Defina uma classe `Programa` com o método `main` e crie objetos da classe `Pessoa`, imprimindo suas informações na tela.

4. Escreva uma classe para representar uma Pilha. Uma pilha é uma estrutura de dados em que objetos são armazenados seguindo a regra de que, ao retirar objetos, esses devem ser removidos na ordem inversa a que foram adicionados.

A classe deverá possuir a seguinte interface:

```
boolean adicionar( Object o );  
Object retirar();  
boolean estáVazia();
```

O método adicionar retorna verdadeiro caso o elemento tenha sido adicionado com sucesso a fila.

Defina uma classe Programa com o método main e crie uma pilha. Depois adicione e retire objetos nessa pilha, imprimindo as informações dos objetos retirados na tela.

5. Um vetor é um ente matemático que associa a um determinado ponto no espaço um valor (módulo), direção e sentido. São utilizados na física para representar conceitos como força e aceleração.

Dados dois vetores tridimensionais  $A = a_1x + a_2y + a_3z$  e  $B = b_1x + b_2y + b_3z$ , algumas operações definidas para os vetores estão listadas abaixo:

**Adição:**  $A + B = (a_1 + b_1)x + (a_2 + b_2)y + (a_3 + b_3)z$

**Multiplicação por escalar:**  $\text{Num} * A = \text{Num} * a_1x + \text{Num} * a_2y + \text{Num} * a_3z$

**Produto escalar:**  $A \cdot B = a_1b_1 + a_2b_2 + a_3b_3$

Escreva uma classe “Vetor” para representar um vetor tridimensional. A interface da classe deverá ser formada pelos seguintes métodos:

- Um construtor;
- Vetor adicionar( Vetor v );
- void multiplicação\_Escalar(double num);
- double produtoEscalar( Vetor v);

6. Um número complexo é todo aquele escrito na forma  $Z = a + b*i$ , onde  $a$  e  $b$  são números reais e  $i = (-1)^{1/2}$ . Dados dois números complexos  $z = a + b*i$  e  $w = c + d*i$ , algumas operações definidas para números complexos estão listadas abaixo:

Adição:  $z + w = (a + bi) + (c + di) = (a + c) + (b + d)*i$

Subtração:  $z - w = (a + bi) - (c + di) = (a - c) + (b - d)*i$

Multiplicação:  $z.w = (a+bi).(c+di) = (ac-bd) + (ad+bc)*i$

Escreva uma classe para representar um número complexo. A classe deverá utilizar encapsulamento, conter um construtor e possuir a seguinte interface:

```
NumeroComplexo adicionar( NumeroComplexo c );  
NumeroComplexo subtrair(NumeroComplexo c);  
NumeroComplexo multiplicar(NumeroComplexo c);
```