

Sistemas Distribuídos



ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Ficha de Recuperação: POO usando Java

Parte 1

Exercício 1

Considere a seguinte matriz:

11	7	333
-20	-23	63
-22	501	10000

Escreva um programa que com base na matriz apresentada, apresente a soma e a média de todos os elementos da matriz.

Exercício 2

Considere a seguinte declaração de um vetor de inteiros.

```
int[] lista = {12, 5, -21, 10, -345, 22, 50, -125, 80, -1};
```

Escreva um programa que permita:

- Calcular o resultado da multiplicação de todos os elementos positivos;
- Identificar quantos elementos são negativos;
- Identificar qual o maior número.

Exercício 3

Considere a seguinte declaração de um vetor de caracteres que representa o nome e apelido de uma pessoa:

```
char[] nome = {'A', 'n', 'a', ' ', 'S', 'a', 'n', 't', 'o', 's',  
'\n'};
```

Escreva um programa que permita imprimir:

- O nome e o apelido no formato: Apelido, Nome;
- O número de vogais;
- O número de consoantes.

Exercício 4

Escreva um programa que através da sua invocação a partir da linha de comandos, receba dois parâmetros correspondentes ao primeiro e último nome de uma pessoa e apresente no ecrã esse nome no seguinte formato: apelido, nome.

Exercício 5

Considere a declaração dos seguintes vetores de inteiros:

```
int[] listaA = {2, -5, -121, 102, -35, -2, 0, -125, 802, -10};  
int[] listaB = {6, 99, -1, 12, 1, -2};
```

Escreva um programa que permita:

- Unir os dois vetores num novo vetor. Imprima o vetor resultante;
- Apresente quantos elementos repetidos existem no vetor criado na alínea a;
- Preencher um novo vetor com os elementos do vetor: "listaA" que não se encontram no vetor: "listaB";
- Preencher um novo vetor com os elementos que se encontram simultaneamente nos dois vetores (sem repetidos).

Parte 2

Desenvolva uma aplicação que permita gerir pizzas de uma pequena pizzaria. Cada pizza disponibilizada pelo restaurante é composta por um conjunto de ingredientes associados (no máximo cada pizza terá 5 ingredientes) e respetivas quantidades.

Cada ingrediente é identificado pelo seu código, nome, unidade de medida (por exemplo: Gramas, Litros ou Unidades) e o número de calorias associadas. Cada pizza é caracterizada pelo seu código, nome, descrição, preço, tamanho (Pequena, Média ou Grande), número de ingredientes e uma coleção de ingredientes que representam a composição da pizza.

Exercício 1

1.1. Num *package* `pp_fp01.PizzaRestaurant`, crie as classes necessárias para responder aos requisitos do problema, considerando que:

- Deve garantir o encapsulamento de todas as classes criadas;
- Deve criar métodos de acesso necessários para todas as classes criadas;
- Num *package* `pp_fp01.PizzaRestaurant.Enums`, deve criar as enumerações para representar a unidade do ingrediente (Gramas, Litros ou unidades) e para o tamanho da pizza (Pequena, Média ou Grande). A impressão da unidade dos ingredientes e do tamanho da pizza deve ser apresentada com uma mensagem descritiva (exemplo na Figura 1);

```
public enum PizzaSize {  
    SMALL, MEDIUM, BIG, KING;  
  
    public static String PizzaSizeToString(PizzaSize size) {  
        switch (size) {  
            case SMALL:  
                return "The pizza is small.";  
            case MEDIUM:  
                return "The pizza is medium.";  
            case BIG:  
                return "The pizza is big.";  
            case KING:  
                return "The pizza is king size.";  
            default:  
                return "The pizza is medium.";  
        }  
    }  
}
```

Figura 1: Exemplo de enumeração

1.2. Teste as classes criadas através de uma classe `PizzaDemo`. Crie no mínimo, duas pizzas (deverá utilizar uma coleção) com pelo menos três ingredientes cada.

1.3. Na classe `Pizza`, adicione métodos que permitam:

- Associar novos ingredientes a uma Pizza até a um máximo de 5. Não deve ser permitido aceder diretamente à variável que representa a coleção dos ingredientes da pizza;
- Editar a quantidade de um ingrediente que pertença à coleção de ingredientes de uma pizza;
- Remover um ingrediente (identificando o ingrediente pelo seu id) que pertença à coleção de ingredientes;
- Determinar o número calorias da Pizza;
- Apresentar uma descrição detalhada da pizza, assim como dos seus ingredientes.

Teste as alterações na classe `PizzaDemo`.