

**ESTGF**ESCOLA SUPERIOR DE TECNOLOGIA  
E GESTÃO DE FELGUEIRAS

IPP – Instituto Politécnico do Porto

**LEI – Licenciatura em Engenharia Informática****PP – Paradigmas de Programação**

2º Semestre ■ Docentes: BMO, RJS, TSM e Dcarneiro

Ficha Prática 6

**Documentação complementar:**

- [Classes](#)
- [Objetos](#)
- [Informação adicional](#)

## Parte 1

Desenvolva uma aplicação que permita gerir pizzas de uma pequena pizzeria. Cada pizza disponibilizada pelo restaurante é composta por um conjunto de ingredientes associados (no máximo cada pizza terá 5 ingredientes) e respectivas 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_fp06.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_fp06.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);
- Utilize a palavra reservada `this` para se referir a cada variável de instância em todas as classes criadas.

```
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**

## Resolução parcial:

```
public class Pizza {
    private final int MAX_INGREDIENTS = 5;
    private int id;
    private String name;
    private String description;
    private float price;
    private PizzaSize size;
    private PizzaIngredients[] ingredients;
    private int numberOfIngredients;

    /**
     * Método construtor para a criação de uma instância de
     * {@link Pizza pizza}.
     *
     * @param id Código que identifica uma <b>Pizza</b>
     * @param name Nome da <b>Pizza</b>
     * @param description Descrição resumida da <b>Pizza</b>
     * @param price Preço da <b>Pizza</b>
     * @param size Tamanho associado à <b>Pizza</b>
     */
    public Pizza(int id, String name, String description, float price, PizzaSize size) {
        ingredients = new PizzaIngredients[MAX_INGREDIENTS];
        numberOfIngredients = 0;
        this.id = id;
        this.name = name;
        this.description = description;
        this.price = price;
        this.size = size;
    }
}
```

**Figura 2:** Excerto da classe Pizza

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`.

## Parte 2

Tendo por base a Parte 1 da ficha prática 5, crie um package `pp_fp06.Expenses`.

### Exercício 1

1.1. Tenha em consideração as classes: `Date`, `Expense` e `User` do projeto anterior e realize as seguintes alterações tendo em consideração que :

- Deve garantir o encapsulamento de todas as classes;
- Deve criar métodos de acesso necessários para todas as classes;
- Utilize a palavra reservada `this` para se referir a cada variável de instância em cada uma das classes criadas;
- Crie enumerações para suportar os vários tipos de despesa, assim como os vários tipos de moeda (`ExpenseType` e `CurrencyType` respetivamente);
- Utilize uma coleção para armazenar todas as despesas.

1.2. Altere a classe `ExpensesDemo` de forma a testar as alterações introduzidas no exercício. A impressão do conteúdo das enumerações deverá ser realizado com uma mensagem descritiva.

1.3. Adicione um atributo género (Masculino ou Feminino) para o utilizador. Crie uma enumeração para representar este novo atributo, assegurando que a impressão do género é realizada com uma mensagem descritiva;

1.4. Implemente métodos na classe `User` para:

- Associar uma nova despesa a um utilizador até a um máximo de 30 despesas. Garanta ainda que não existem despesas com o mesmo número;
- Remover uma despesa (identificando a despesa pelo id);
- Remover todas as despesas de um mês;
- Alterar a descrição de uma despesa (identificando a despesa pelo ID) que pertença à lista de despesas do cliente.

1.5. Na classe `ExpensesDemo` teste as alterações realizadas;

**Gere o Javadoc para o projecto utilizado na resolução desta ficha de trabalho.**