

Sumário

- Métodos
- Sobrecarga de Métodos
- Encapsulamento
- Métodos de acesso
- Modificadores de acesso

Documentação complementar Java:

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

Exercício 1

Pretende-se o desenvolvimento de uma aplicação que dê suporte à gestão de ementas por uma pequena pizzaria.

Considere que a ementa é constituída pelos seguintes atributos

- Designação
- Descrição
- Código de identificação
- Data de início (data em que a ementa entra em vigor)
- Data de fim (não necessita de ser pré-estabelecida)
- Número de pizzas que fazem parte da ementa
- Conjunto de pizzas que constituem a ementa

Cada pizza é constituída por:

- Código
- Nome
- Descrição
- Preço
- Tamanho (*Pequena, Média, Familiar*)
- Número de ingredientes
- Coleção de ingredientes (no máximo 5)

Por sua vez, um ingrediente terá:

- Código
- Nome
- Origem (animal, vegetal ou mineral)
- Número de calorias

1.1) Num `package pp_fp06.pizza_restaurant` crie as classes necessárias para responder aos requisitos do problema considerando que deve

- Garantir o encapsulamento de todas as classes criadas;
- Criar os métodos de acesso necessários para as classes criadas;
- Criar as enumerações (no `package pp_fp06.pizza_restaurant.enums`) para representar o tamanho da `Pizza` e de outros atributos que considere necessários
- Implementar métodos para “imprimir” uma descrição para os valores de cada enumeração (tome como exemplo a Figura 1)
- Utilizar a palavra reservada `this` para se referir a cada variável de instância em todas as classes criadas.

```
/** Enumerator that defines the possible sizes for ...4 lines */
public enum PizzaSize {
    /** Small size ...3 lines */
    SMALL,
    /** Medium size ...3 lines */
    MEDIUM,
    /** Big size ...3 lines */
    BIG;

    /** Convert the enumerator size value into human readable format ...5 lines */
    public static String pizzaToString(PizzaSize size) {
        switch (size) {
            case SMALL:
                return "The pizza is small.";
            case BIG:
                return "The pizza is big.";
            default:
                return "The pizza is medium.";
        }
    }
}
```

Figura 1 - Exemplo de enumeração

1.2) Teste as classes criadas anteriormente através de uma classe `PizzaDemo`. Deverá criar, no mínimo, duas pizzas com, pelo menos, três ingredientes cada.

1.3) Na classe `Pizza` adicione métodos que permitam:

- Adicionar novos ingredientes a uma `Pizza`, até um máximo de 5 ingredientes;
- Remover um ingrediente de uma `Pizza`;
- Obter informação de um `Ingrediente` dado o seu id;
- Apresentar uma descrição detalhada da `pizza`, incluindo os seus ingredientes;
- Determinar o número de calorias da `Pizza`.

Para adicionar, encontrar e remover ingredientes devem existir métodos específicos para manipular a coleção de ingredientes. Tenha em atenção que não deve ser permitido o acesso direto à variável que representa a coleção de ingredientes da `pizza` por outros métodos da classe e usar os métodos adicionar, encontrar e remover previamente criados.

Exercício 2

2.1) Altere a classe `Ementa` de forma a permitir associar um máximo de 10 `Pizzas`.

2.2) Realize as modificações necessárias para que seja possível armazenar várias `Ementas`. Considere que apenas poderá existir uma `ementa` ativa num determinado momento (crie/edite o(s) método(s) necessário(s) para o efeito).

Gere o JavaDoc para o(s) projeto(s) utilizado(s) na resolução desta ficha de trabalho.