

## Ficha de exercícios 4

**Ex. 1:** Crie um novo projeto. Numa loja, toda a fruta vendida tem um nome e um preço base. Porém, a loja vende fruta em três modalidades: à unidade, ao peso ou pelo volume. Crie uma hierarquia onde Fruta é uma classe abstrata, com as subclasses FrutaUnidade, FrutaPeso, e FrutaVolume.

- a) A classe abstrata deverá ter as variáveis comuns a todas as subclasses, bem como o método abstrato `double pagar()`. Inclua os métodos concretos que considerar adequados.
- b) Defina as classes FrutaUnidade, FrutaPeso, e FrutaVolume, como subclasses de Fruta, dando a adequada implementação ao método `double pagar()` para cada caso.

Escreva a função `main()` para testar as classes acima descritas.

**Ex. 2:** Defina a classe Cesto, que representa um conjunto de frutas adquiridas. Esta classe deverá implementar o seguinte:

- a) Uma variável de instância para o total de itens no cesto e uma constante que determina a capacidade máxima do cesto como sendo 100. Inclua também um array de Fruta para representar o cesto, que é um conjunto de frutas.
- b) Um método que insere no cesto diferentes tipos de fruta.
- c) Um método que calcula o valor total da fruta no cesto.
- d) Um método que determina o número de frutos de um dado tipo existentes no cesto.
- e) Um método que determina o valor total gasto em frutos de um dado tipo.

Na classe principal, crie um cesto e insira vários tipos de fruta. Teste todos os métodos.

**Ex. 3:** Toda a fruta a peso e toda a fruta a volume podem ter um desconto de uma determinada percentagem. Crie uma interface Descontavel que tem um método `double descontar(double percentagem)` que será implementado pela classe FrutaPeso e pela class FrutaVolume. Faça as alterações necessárias nestas classes. Escreva a função `main()` para testar as respetivas alterações.