



Escola de Engenharia da Universidade do Minho
Mestrado Integrado em Eng. Electrónica Industrial e Computadores

2014/2015

MIEEIC

(1º Ano)
2º Sem

Complementos de Programação de Computadores

Luís Paulo Reis

Aula Prática 4: Exercícios de Introdução à Programação Orientada a Objectos

Objectivos:

Esta Folha de Exercícios destina-se a:

- Compreender os conceitos de classes e encapsulamento, dados públicos e dados privados.
- Compreender os conceitos de operador e sobrecarga de operadores.
- Compreender a utilização das classes string e vector.

Os exercícios aqui propostos deverão ser realizados no mais simples ambiente de desenvolvimento possível para a linguagem C: editor de texto de programação ou editor DevC++ e ferramentas da GCC (GNU Compiler Collection) e afins.

Exercício 4

Suponha o seguinte código representando a interface de um Jardim que se encontra contido num ficheiro `Jardim.h` e se destina a gerir a informação sobre todas as plantas de um Jardim/Horto. O nome das funções e das variáveis indica a sua funcionalidade no programa. Na rotina `'main()'` do programa de gestão de informação do Jardim, entre outras coisas, consta a declaração (e definição) `'CJardim jard;'`. Suponha que foram incluídos todos os `#include` necessários. Pretende-se criar o ficheiro `Jardim.cpp` contendo a implementação da classe e o ficheiro `main_jardim.cpp` contendo código que permita testar as funcionalidades da classe.

```
1 // #includes e declarações gerais
2 const int MINVALOR = 0;
3 const int MAXVALOR = 10000;
4
5 class CPlanta {
6     protected:
7         string nome;
8         string especie;
9         int valor, custo;    // em euros
10    public:
11        CPlanta(string, string, int, int);
12        string getEspecie() const;
13        string getNome() const;
14        int getValor() const;
15        int getCusto() const;
16    };
17
18 class CJardim{
19     vector <CPlanta> plantas;
20    public:
21        void adicionarPlanta(string, string, int, int);
22        void imprimirJardim(ostream &);
23        void ordenarJardim();
24    };
```

4.1) Escreva o código da função `int getValor() const;` que retorna o valor de uma dada planta.

4.2) Escreva o código da função `'void adicionarPlanta(string nom, string esp, int val, int cus)'`, que acrescenta à estrutura de dados a informação sobre uma nova planta adquirida.

4.3) Escreva o código de `'imprimirJardim(ostream &os)'`, função que, se invocada em `'main()'` como `'jardim.imprimirJardim(cout);'`, vai permitir mostrar no ecrã do computador a lista de plantas de `'jardim'`, mostrando para cada uma a sua espécie, nome, valor e custo.

4.4) Escreva o código da função `'void ordenarJardim()'`, que altera a estrutura de dados com a informação das plantas, ordenando-a internamente segundo o valor das plantas, da menos para a mais valiosa.

4.5) Implemente um novo operador `==` que compara duas plantas retornando verdadeiro caso tenham o mesmo nome, espécie, valor e custo

4.6) Escreva o código de uma nova função membro da classe `CJardim`, `gravarJardim` que receba o nome de um ficheiro (por exemplo `"jardimEstrela.txt"`) e grave o conteúdo do jardim ordenado segundo o valor das plantas e não incluindo plantas repetidas. Sugestão: use as funções e operadores das alíneas 2.4 e 2.5 (mesmo que não os tenha implementado)

4.7) Construa uma função membro da classe `CJardim` designada: `void substJardim(string nomeParte, string nomeSubst);` que substitua no nome de todas as plantas do jardim que incluam `nomeParte`, essa parte do nome por `nomeSubst`. Exemplo `substJardim("Brasil", "América")` transforma `"Cana do Brasil"` em `"Cana da América"`, `"Planta Brasil Nova"` em `"Planta América Nova"` e assim por diante.

4.8) Construa a função `main` que lhe permita testar todas as funcionalidades da classe `CJardim` implementada.