



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 3c: Exercícios de Introdução à Programação Orientada a Objectos

Objectivos:

Esta Folha de Exercícios destina-se a:

- Compreender os conceitos de classes e encapsulamento, dados público e dados privados.
- Compreender os conceitos de operador e sobrecarga de operadores.

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 3c)

Pretende-se escrever um programa em C++, denominado "CentroComercial", para gerir um centro comercial. O programa deve conter uma classe denominada `CCentroComercial` com a seguinte estrutura:

```
class LOJA { //Classe simples podia ser uma struct
public:
    string nome;
    string tipo;
    int renda;
};

class CCentroComercial {
private:
    vector<LOJA> lojas;

public:
    void ler_lojas(const char *nome);
    void imprimir_lojas(ostream &output, string tipo);
    void imprimir_lojas(ostream &output);
    void ordenarLojas ();
};
```

Execute as tarefas seguintes, tendo sempre o cuidado de testar no programa principal (função `main`), todas as funções criadas. O código fonte do programa deve ser escrito em ficheiros com o nome "main_CentroComercial.cpp", "CentroComercial.h" e "CentroComercial.cpp".

a) Implemente o membro-função:

```
void CCentroComercial::ler_lojas(const char * filename)
```

que lê a informação de um conjunto de lojas de um ficheiro de texto para o vector lojas. O ficheiro está organizado em blocos de 3 linhas, contendo o nome da loja, o tipo da loja e a renda (euros/mês), por exemplo:

```
MacDonalds
Fast Food
1000
Zara
Roupa
5000
```

b) Implemente os membros-função

```
void imprimir_lojas(ostream &output, string tipo);
void imprimir_lojas(ostream &output);           //todas
```

que escreve no stream output os elementos do vector lojas, 1 por cada linha. Em cada linha os campos de cada loja deverão ser separados por ponto-e-vírgula.

c) Implemente o membro-função

```
void ordenarLojas ();
```

que ordene o vector lojas por renda através do método ordenação por selecção.

d) Implemente o operador

```
ostream& operator<< (ostream& os, const CCentroComercial::LOJA & loja);
```

Altere os membros-função `imprimir_lojas()` para usar este operador.