

 Universidade do Minho	Escola de Engenharia da Universidade do Minho Mestrado Integrado em Eng. Electrónica Industrial e Computadores Complementos de Programação de Computadores	2014/2015 MIEEIC (1º Ano) 2º Sem
Luís Paulo Reis		
Aula Prática 3e: 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 3e

Pretende-se escrever um programa em C++, denominado “Videoteca”, para gerir uma videoteca particular. O programa deve conter uma classe CVideo com a seguinte estrutura:

```
class CVideo
{
    string nome;
    string realizador;
    int ano;
public:
    CVideo (string nome, string realizador, string ano);
    int getAno ();
    string getNome();
    string getRealizador();
};
```

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 nos ficheiro com o nome “main_Videoteca.cpp”, “Videoteca.h” e “Videoteca.cpp”.

a) Implemente a função

```
int lerVideoteca (vector<CVideo *>& vt, const char * filename);
```

para ler de um ficheiro a informação sobre os nossos filmes.

O ficheiro está organizado em blocos de 3 linhas, contendo o nome do filme, o realizador e o ano:

```
O Bom Pastor
Robert de Niro
2006
Ghost Rider
```

b) Implemente os membros-função `getAno()`, `getNome()` e `getRealizador()` da classe `CVideo` e a função global

```
void mostraVideos (vector<CVideo *>& vt, int minAno, int maxAno);
```

que imprime na consola todos os videos realizados entre os anos `minAno` e `maxAno`.

c) Implemente o operador

```
bool CVideo::operator< (CVideo& v2)
```

que retorna o resultado da comparação dos anos e a função global

```
int comparaPorAno (const CVideo &v1, const CVideo &v2)
```

que retorna `-1` se o ano de `v1` é inferior ao ano de `v2`, `0` se o ano de `v1` é igual ao ano de `v2` e `1` se o ano de `v1` é superior ao ano de `v2`.

d) Implemente a função

```
int removeVideos (vector<CVideo *>& vt, int minAno, int maxAno)
```

que remove todos os vídeos realizados entre os anos `minAno` e `maxAno`, inclusivé.