



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 3f: 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 3f

Pretende-se escrever um programa em C++, denominado “medidas”, que processa um conjunto de medidas (números inteiros). O programa deve conter uma classe denominada CMedidas com a seguinte estrutura:

```
class CMedidas {  
    int nmed;  
    vector <int> medidas;  
  
public:  
    void ler_medidas(const char *nome);  
    void imprimir_medidas(ostream &output);  
    int pesquisa_seq(const vector<int> &medPesq);  
    void bubble_sort(int ordem);  
};
```

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 três ficheiros com os nomes “main_medidas.cpp”, “medidas.h” e “medidas.cpp”.

a) Implemente a função-membro:

```
void CMedidas::ler_medidas (const char *nome)
```

que lê um conjunto de números inteiros de um ficheiro de texto para o vector medidas e coloca em nmed o número total de medidas. Caso ocorra algum erro na leitura nmed deve ficar com o valor -1. Construa um ficheiro, designado “medidas.txt” com os seguintes valores para realizar o teste do programa:

```
9 10 3 5 0 1 2 3 4 5 6 7 8 20 0 15 3 4 5 10 0 3 4 5 3 0 23 14 0
```

b) Implemente a função-membro:

```
void CMedidas::imprimir_medidas(ostream &output)
```

que escreve no stream output os elementos do vector medidas, separados por um espaço – a medida de valor zero indica mudança de linha.

c) Construa a função-membro:

```
void CMedidas::bubble_sort(int ordem)
```

que ordene o vector por ordem crescente (caso ordem seja um inteiro positivo) ou decrescente (caso ordem seja um inteiro negativo), utilizando o método “bubble_sort”. Caso ordem seja 0, o vector deve permanecer inalterado.

d) Implemente a função-membro:

```
int CMedidas::pesquisa_bin(const vector<int> &medPesq)
```

que pesquise, utilizando pesquisa binária, no vector de medidas, a posição da primeira ocorrência da sequência de elementos medPesq. A função deve retornar -1 caso a sequência

não seja encontrada.