



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 3g: 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 3g

Pretende-se escrever um programa em C++, denominado "JardimZoologico", para gestão de um Jardim Zoológico. O programa deve conter uma classe denominada CJardimZoologico com a seguinte estrutura:

```
class ANIMAL {
public:
    string categoria; // urso, girafa, etc
    string nome;
    string veterinario;
    int idade;        // anos
};

class CJardimZoologico {
private:
    vector<ANIMAL> animais;

public:
    bool animalJovem(const char *nome);
    vector<ANIMAL> animaisJovens();
    void loadAnimals(istream& is);
    vector<string> veterinarioComMaisTrabalho();
};
```

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_JardimZoologico.cpp", "JardimZoologico.h" e "JardimZoologico.cpp".

a) Implemente o membro-função

```
bool CJardimZoologico::animalJovem(const char *nome);
```

que verifica se um dado animal é jovem e

```
vector<ANIMAL> CJardimZoologico::animaisJovens();
```

que retorna o conjunto de todos os animais jovens. Um animal é jovem se não tiver mais de 5 anos.

b) Implemente a função

```
void CJardimZoologico::loadAnimals(istream& is);
```

que carrega a informação sobre os animais existente num ficheiro. Esta deve ir por linhas da seguinte forma:

```
urso
Mikas
João Carvalho
5
girafa
Margarida
Tiago Santos
8
```

c) Implemente o operador

```
bool CJardimZoologico::operator> (CJardimZoologico& zoo2);
```

para comparar dois jardins zoológicos. Um jardim zoológico é maior que um segundo se a soma das idades dos seus animais for superior à soma das idades dos animais no segundo.

d) Pretende-se saber qual é o veterinário que mais animais cuida. Implemente o membro-função

```
vector<string> CJardimZoologico::veterinarioComMaisTrabalho();
```

que devolve o(s) veterinário(s) que mais animais cuida(m).