

2ª aula prática – Herança e subclasses; Polimorfismo; Membros Estáticos; Documentação (Doxygen)

Faça download do ficheiro *aeda2021_p02_extra.zip* da página da disciplina e descomprima-o (contém a pasta *lib*, a pasta *Tests* com os ficheiros *zoo.h*, *zoo.cpp*, *animal.h*, *animal.cpp*, *veterinary.h*, *veterinary.cpp* e *vets.txt*, e os ficheiros *CMakeLists* e *main.cpp*)

- Note que os testes unitários deste projeto estão comentados. Retire os comentários à medida que vai implementando os testes.
- Deverá realizar esta ficha respeitando a ordem das alíneas.

Enunciado

Pretende-se escrever um programa em C++ para gestão de um Jardim Zoológico. Por uma questão de simplificação, admita que o jardim zoológico só possui cães e morcegos. O programa deve conter as classes a seguir descritas parcialmente:

```
class Animal {
protected:
    string name;
    int age;
    Veterinary *vet;
    static int youngest;
public:
    // returns a string with name, age, veterinary info
    virtual string getInfo() const;
    virtual bool isYoung() const = 0;
    static int getYoungest();
};
```

```
class Zoo {
    vector<Animal *> animals;
    vector<Veterinary *> veterinarians;
public:
    void addAnimal(Animal *a1);
    void allocateVeterinarians(istream &isV);
    string getInfo() const;
    bool isYoung(string nameA);
};
```

```
class Flying {
    int maxVelocity;
    int maxAltitude;
public:
    //returns also maximim_velocity and maximum_altitude
    virtual string getInfo() const;
};
```

```
class Dog: public Animal {
    string breed;
public:
    //...
```

```
class Bat: public Animal, public Flying {
    // ...
};
```

```
class Veterinary {
    string name;
    long codOrder;
    //...
};
```

a)

- Implemente os construtores de todas as classes.
- Implemente também o **membro função estático**

```
int Animal::getYoungest()
```

que retorna a idade do mais jovem dos animais. Atualize convenientemente o membro estático `Animal::youngest`
- Implemente também, nas classes respetivas, o membro função: `bool isYoung () const`
Considere que um cão é jovem se tiver menos de 5 anos, e um morcego é jovem se tiver menos de 4 anos.

b) Implemente o membro-função

```
void Zoo::addAnimal(Animal *a1)
```

que adiciona o animal **a1* ao conjunto de animais do zoo (vetor *animais*).

c) Implemente o membro-função

```
string Zoo::getInfo () const
```

que retorna numa *string* a informação sobre todos os animais do Zoo. A informação sobre um animal é o conjunto dos valores de todos os seus membros dado (incluindo o veterinário).

Nota: este teste não falha, deve verificar a escrita correta da informação.

d) Implemente o membro-função

```
bool Zoo::isYoung(string nameA)
```

que verifica se o animal de nome *nameA* existe e é jovem.

e) Implemente o membro-função

```
void Zoo::allocateVeterinarians(istream &isV)
```

Este método carrega a informação sobre os veterinários existentes no ficheiro *isV* que contém a informação sobre cada veterinário em duas linhas: nome e código da ordem (para teste, use o ficheiro *vets.txt*). Os veterinários devem ser distribuídos uniformemente para os animais do zoológico.

Nota: Para aceder a ficheiros no seu programa (necessário ler o ficheiro *vets.txt*), pode:

- a) especificar o caminho absoluto, ou
- b) alterar “Working directory” no CLion (Run -> Edit Configurations... -> Working directory) para a pasta onde os ficheiros se encontram, ou
- c) adicionar ao ficheiro CMakeLists.txt a diretiva

```
set(CMAKE_RUNTIME_OUTPUT_DIRECTORY "${CMAKE_CURRENT_SOURCE_DIR}/Tests")
```

neste caso, os ficheiros compilados e a ler/escrever são colocados na pasta *projeto/Tests*

f) Adicione o membro-função

```
bool Zoo::removeVeterinary (string nameV)
```

que remove o veterinário de nome *nameV* do vetor de veterinários do Zoo. Se algum animal tiver este veterinário, deve passar para o veterinário seguinte do vetor *veterinarios*. Se não existir nenhum veterinário de nome *nameV*, esta função retorna *false*.

g) Implemente o operador

```
bool Zoo::operator < (Zoo& zoo2) const
```

para comparar dois jardins zoológicos. Um jardim zoológico é menor que um segundo se a soma das idades dos seus animais for inferior à soma das idades dos animais do segundo zoo.

h) Efetue a documentação dos membros-função implementados (use Doxygen).