Umut Cem Avin
150140018
21.04.2018

BLG 252 - Object Oriented Programming
Homework 2 Report

A) Introduction

In this homework a C++ program is written which simulates mutation of species in danger of extinct. Object Oriented Programming approaches have been used by considering Inheritance and Polymorphism.

B) Implementation

An abstract base class Grayling is constructed and different variations such as Grayling1, Grayling2, Grayling3 classes is constructed and derived from it. “grayling.h” file includes the definitions of these methods and also attributes for each grayling type.

As you can see below, methods are public members and attributes are protected member of the class. So that, I can reach the methods from anywhere. The important point here is using protected members for attributes. With this usage, base class’s protected values can be accessible in classes that inherit from that class.

In the code, “const” is used to prevent unnecessary or accidental changes.

```
14 class Grayling{
15 public:
16     //Methods
17     Grayling(){isAlive=0;}; //Default Constructor
18     Grayling(char, string); //Constructor
19     Grayling(const Grayling &g2, char, string); //Copy Constructor
20     /*
21      These functions are virtual so that we can make certain operations in
22      our base class for other Grayling classes.
23     */
24     virtual void print() const;
25     virtual void givebirth();
26     virtual void aging();
27     virtual ~Grayling() {};
28 protected:
29     //Attributes
30     short age;
31     string name;
32     char gender;
33     bool isAlive;
34     bool isMutant;
35     string motateto;
36     double MPI;
37     double MP;
38     double nextMP;
39     int offsCounter; // Counts offsprings
40     int deathAge;   // Grayling's death age
41     int maxOffspring; // Grayling's possible offspring number
42 };
```

Figure 1

”grayling.cpp” file includes print(), givebirth() and aging() functions for all classes. In the functions, graylings are separated by considering their attributes’ values and required operations has been done according to that. A function sample is given below.

```
void Grayling::givebirth(){
    /*
     Conditional statements to differ graylings by considering their life status
     gender and offspring number. These values show us their availability for
     giving birth.
    */
    if(isAlive == 0){
        if(offsCounter >=maxOffspring)
            cout << "- Trying to GIVE BIRTH BUT " << name <<
            " is not alive and no ability to give birth!" << endl;
        else
            cout << "- Trying to GIVE BIRTH BUT " << name <<
            " is not alive!" << endl;
    }
    else if(gender == 'f' && isAlive == 1){
        if(offsCounter < maxOffspring){
            cout << "- " << name << " gave birth to " << offsCounter+1 <<
            " offsprings!" << endl;
            offsCounter++;
        }
        else if(offsCounter >=maxOffspring){
            cout << "- Trying to GIVE BIRTH BUT " << name <<
            " has no ability to give birth!" << endl;
        }
    }
    else if(gender == 'm' && isAlive == 1)
        cout << "- Trying to GIVE BIRTH BUT " << name <<
        " has no ability to give birth!" << endl;
}

/*
```

Figure 2

IMPORTANT

At the end of the main.cpp function, a “getch();” command is given. Since, I used a MacOSX system in this project, I didn’t use this command because “conio.h” header file which includes getch(), doesn’t work on OSX operating systems.

C) Result

Compile command: g++ main.cpp -o hw2

SSH output for this program is given below.

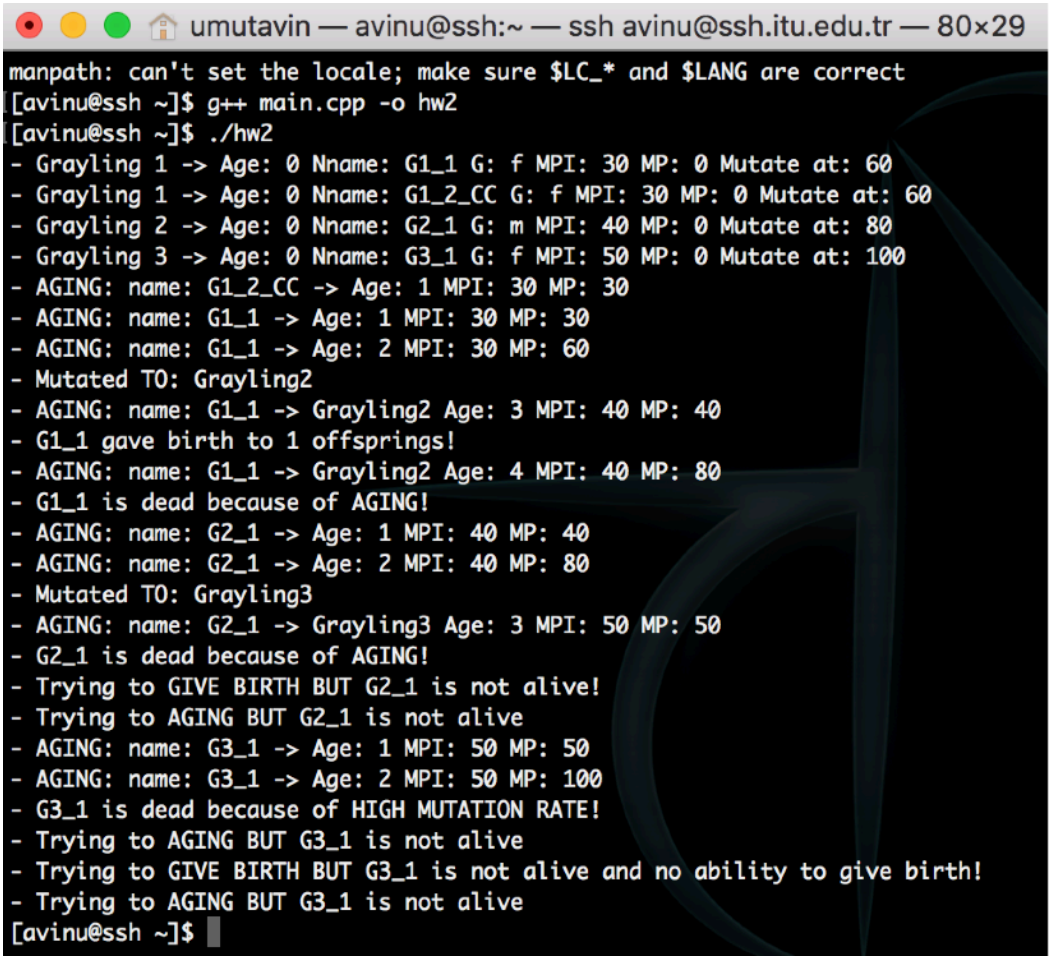

Figure 3