Ian Sabey

Unit 3 Submission Node 1

CS288 – C++ Programming

Prof. Jeffery Sanford

With me having a broader history in Java than C++, I had a slightly difficult time working around the syntactic differences (I test software written in Java daily). I did however enjoy learning the different but not so different coding style. I took the same approach in writing this assignment as the others, writing the main class definition in a header file and pulling it in to the cpp. This one stumped me on whether it would be multiple or single inheritance, in the fact that I, for some reason, couldn’t spark the creative side needed to go in depth with different attributes for the different classes that would be a generalization. Conclusively, the data structure in my assignment ended up with single inheritance, in that, each derived class consisted of one base class as opposed to multiple base classes. I could have made it multiple inheritance by adding another base class Habitat, and returned what environment each derived class would live in.

**Vertebrates.h**

#include <iostream>  
#include <string>  
  
**using namespace** std;  
  
**class** Vertebrate  
{  
**public**:  
 string skinCover, animalType, appendages;  
 //constructor  
 Vertebrate()  
 {  
 skinCover = getSkinCover();  
 animalType = getAnimalType();  
 appendages = getAppendages();  
 };  
 **void** setAppendages(**int** legs) {  
 **int** appendages = legs;  
 };  
 string getSkinCover() {  
 **return** skinCover;  
 };  
 string getAppendages() {  
 **return** appendages;  
 };  
 string getAnimalType() {  
 **return** animalType;  
 };  
};

// here is an example of an addition base class to make the data structure multiple

//class Habitat  
//{  
//public:  
// string livingEnv;  
// Habitat()  
// {  
// livingEnv = getLivingEnv();  
// }  
//  
// string getLivingEnv() {  
// return livingEnv;  
// }  
//};  
//  
//// if i included the above, each derived class below would be restructured to include the Habitat class  
//class Bird: public Vertebrate, public Habitat {  
//public:  
// Bird(string birdType) {  
// skinCover = "feathers";  
// animalType = "birds";  
// livingEnv = "nests, generally in trees";  
// setAppendages(2);  
// cout << birdType << "s are " << animalType << endl;  
// cout << birdType << "s have " << skinCover << endl;  
// cout << birdType << "s have " << appendages << " legs" << endl;  
// cout << birdType << "s (for the most part) live in " << livingEnv << endl;  
// }  
//};

**class** Bird: **public** Vertebrate {  
**public**:  
 Bird(string birdType) {  
 skinCover = "feathers";  
 animalType = "birds";  
 setAppendages(2);  
 cout << birdType << "s are " << animalType << endl;  
 cout << birdType << "s have " << skinCover << endl;  
 cout << birdType << "s have " << appendages << " legs" << endl;  
 }  
};  
  
**class** Amphibian: **public** Vertebrate {  
**public**:  
 Amphibian(string amphType) {  
 skinCover = "mucus";  
 animalType = "amphibians";  
 setAppendages(4);  
 cout << amphType << "s are " << animalType << endl;  
 cout << amphType << "s have " << skinCover << endl;  
 cout << amphType << "s have " << appendages << " legs" << endl;  
 }  
};  
  
**class** Mammal: **public** Vertebrate {  
**public**:  
 Mammal(string mammType) {  
 skinCover = "fur/hair";  
 animalType = "mammals";  
 setAppendages(4);  
 cout << mammType << "s are " << animalType << endl;  
 cout << mammType << "s have " << skinCover << endl;  
 cout << mammType << "s have " << appendages << " legs" << endl;  
 }  
};  
  
**class** Reptile: **public** Vertebrate {  
**public**:  
 Reptile(string repType) {  
 skinCover = "scales";  
 animalType = "reptiles";  
 setAppendages(4);  
 cout << repType << "s are " << animalType << endl;  
 cout << repType << "s have " << skinCover << endl;  
 cout << repType << "s have " << appendages << " legs" << endl;  
 }  
};  
  
//  
// Created by Ian Sabey on 12/24/18.  
//

**main.cpp**

#include <iostream>  
#include <string>  
#include <map>  
#include "Vertebrates.h"  
  
**using namespace** std;  
  
string getUserChoice() {  
 string userResponse, userRespType;  
 cout << "What kind of vertebrate would you like info on? "  
 "birds/mammals/reptiles/amphibians" << endl;  
 cin >> userResponse;  
 **if** (userResponse == "birds") {  
 // secondary question to make the program seem more user friendly  
 cout << "What kind of bird?" << endl;  
 cin >> userRespType;  
 **new** Bird(userRespType);  
 } **else if** (userResponse == "mammals") {  
 // secondary question to make the program seem more user friendly  
 cout << "What kind of mammal?" << endl;  
 cin >> userRespType;  
 **new** Mammal(userRespType);  
 } **else if** (userResponse == "reptiles") {  
 // secondary question to make the program seem more user friendly  
 cout << "What kind of reptile?" << endl;  
 cin >> userRespType;  
 **new** Reptile(userRespType);  
 } **else if** (userResponse == "amphibians") {  
 // secondary question to make the program seem more user friendly  
 cout << "What kind of amphibian?" << endl;  
 cin >> userRespType;  
 **new** Amphibian(userRespType);  
 } **else** {  
 cout << "Please enter an option from above.";  
 getUserChoice();  
 }  
 **return** std::string();  
};  
  
**int** main()  
{  
 getUserChoice();  
}  
  
//  
// Created by Ian Sabey on 12/24/18.  
//