Lab6 - CIT255

Goals

The primary goal of this lab is to write, test and complete a program with multiple classes *and inheritance*. You will also code and use constructors, methods and attributes. This program is an extension of the last lab which adds additional inheritance to your class structure. You will need to re-use your code from the previous lab as a starting point. This will reinforce the need for good design and extending an existing code base. For this lab you will also need to create and efficient design. This is an extension of previous labs with a small change of data. *You must change the previous system to capture all previous data and behavior and add the new data and behavior specified here.*

1. Allows inputs via a simple GUI
2. Instantiates an object from a class you design (class Student or GradStudent) which extends from another class you already designed (class Person).
3. Stores the data in your Student or GradStudent *object*
4. Clear the GUI of all data
5. Functions to restore the data from the Student OR GradStudent object to the GUI
6. Add a Java interface called BirthDateCalculate. You will create one method to take the birth year and calculate the age of any person or student, using this interface, when implemented.

Specifications

The class you create will be a *Person* class. The *Person* class will connect to 2 user-defined classes, *Name* and *Address*. The class will contain the following list of data attributes. You must decide the correct types for each, except the *Name* and *Address* types, as they are additional classes.

You must review all of the required attributes of all defined classes and arrive at an efficient, optimal design that supports inheritance. Then, you must implement that design.

Person class:

- Name (this is a separate class below)
- Address (this is a separate class below)
- Age
- Birth date

Student class:

- Name (this is a separate class below)
- Address (this is a separate class below)
- Major - this must be in a drop down selection from the following majors:
  - CIT
  - Physics
  - Math
  - Plant Science
  - 7-year plan
- Favorite Class - this must be in a drop down selection from the following choices:
  - CIT 255
  - CIT 155
  - Literature of Monty Python
  - Presidency of Kanye West
  - Jazz flute
  - Theory of killer drones
  - no comment
- Age
- Birth date
- Graduation date
- Attended Hogwarts (yes or no)

Name class:

- First Name
- Middle Initial
- Last Name

Address class:

- Address1
- Address2
- City
- State
- ZIP Code (5 digits)

You will also have a child class of Student called GradStudent. Similar to what we have discussed in class. *All of the data in the GUI will be stored in the Student or GradStudent object, not Person.*

GradStudent class:

- Thesis Title
- Advisor Name (this is a separate class - reuse Name class)
- Graduation date
- Major - this must be in a drop down selection from the following majors:
  - CIT
  - Physics
  - Math
  - Plant Science
  - 7-year plan

You must tie the *Person*, *Name* and *Address Student* and *GradStudent* classes together. You will need at least 2 user-defined constructors for each class, based on your design.

You will need to create a GUI that can take as input all of these fields, in the reasonably correct type. Then, you will need to write *get* and *set* functions for each of these. These functions will be the way to message the data between the GUI and the object. On the GUI, there will be three buttons, with the following actions:

- Save - writes the inputted data to the object
- Clear - deletes all of the data from the GUI
- Restore - retrieves the data from the object and sends it back to the GUI

The successful conclusion to this lab will be the saving and restoring of this data, through your newly created object.

Submission

Submit the project file(s) to Blackboard by the required date and time. The grading will be done in lab, but the documents must be submitted to Blackboard as a backup.