Labwork 4 Factory Method and Singleton – Object Orientation with Design Patterns 2015

This lab is worth 5% or 50 points from the total 500 points for labwork this semester

Part 1: Singleton Sun class (15 points)

Create a new project called **Lab4Part1**. Access Moodle and copy the Solar System Singleton classes (all of them) and complete the following task:

Currently this software allows more than ONE Sun to be created in a Solar System. Modify the system so that ONLY ONE instance of the Sun class can be created, i.e., remove the possibility of having more than ONE Sun in any Solar System. Throw an exception if there is an attempt to create more than one instance of the Sun class and catch and handle that exception in the SampleSolarSystem code.

- Modify the Singleton constructor (5 points)
- Implement Singleton exception and catch\handle for one Sun (5 points)
- Test and run program by attempting to create more than 1 Sun (3 points)
- Comments ALL changes\modification made to the program (2 points)

Part 2: Factory Method (Gold, Silver, Bronze based on 1st, 2nd or 3rd)

(15 points)

Create a new project called **Lab4Part2**. Create a JFrame application called **AwardsGUI** with a JPanel (Centered and with a JTextField for text display and a JLabel for an image display) and two JComboBox's (on the South, with choice First(1), Second(2) or Third(3) AND the second Combo to list the awarding body "Atheletics" or "HorseShow"). Create a generic **AwardingBody** class and two subclasses called **AthleticsAwardingBody** and **HorseShowAwardingBody**. The generic **AwardingBody** class should return a String message to all winners "Congratulations you are a winner", this class should also contain the factory method **getPrize(int finishingPoisition)** to be implemented by the subclasses. Create a generic prize class called **Prize** and two subclasses of **Prize** called **Medal** and **Ribbon**. The subclasses of Prize have attributes in the form of images of the prizes that can be awarded, e.g., a picture of a gold, silver or bronze medal, or a red, yellow or blue ribbon etc.

- JFrame created and working (2 points)
- Implement generic\abstract AwardingBody class (3 points)
- Implement ALL subclasses (3 each) and factory methods (6 points)
- Implement Product subclasses (Prize, etc.) (2 points)
- (Test) Return the correct prize for selections at JComboBox (2 points)

Part 3: Factory method to return a correct JFrame based on the selection of the user passed to the Factory Method

(20 points)

Problem: You want to display one of two possible welcome-GUI's in your applications to the user based on the user (new user or returning user) and based on the nature of the application (e.g. secured or insecured). A greeting will be different if the application is secured (login\register options) or non-secured (could use load different greeting for first time visitors or returning visitors)

Possible solution: Use the factory method pattern to return the correct JFrame for a secured application based on the selection from the JComboBox received as a parameter (new user or returning user)

Create a new project called **Lab4Part3**. Create a JFrame called **GreetingDisplay** application which will welcome a user differently in a secured application based on whether they are a new user or an existing user by applying the **factory method** design pattern. Include a JComboBox. The choices on the JComboBox should be “Returning User” or “New User”. Create TWO separate subclasses of JFrame called **LoginFrame**, **RegisterFrame** each of which will return a complete JFrame GUI with fields for the correct entries sought for login or register. **LoginFrame** seeks only username and password in a GUI by returning a JFrame with two fields. **RegisterFrame** includes name, address1, address2, email fields by returning a JFrame with at least the four fields listed. Create a class called **AppWelcome** with a **factory method** called **getGUI(String userType)** to return a complete JFrame of the correct form. Create a subclass of the **AppWelcome** class called **SecuredApp** class which will provide a concrete implementation of the **factory method getGUI(String userType)** and will return the correct GUI for secured apps [Note: It would be possible to implement a similar factory method implementation for non-secured applications which would provide an appropriate GUI for welcoming user appropriately based on whether they are first time visitors (“Welcome”) or returning visitors (“Welcome back”), however, this is not required for this task; one implemented concrete factory method is all that is required].

Required tasks and marking guideline:

• Main application GUI with JComboBox and listeners	(2 points)
• AppWelcome generic class with factory method	(2 points)
• SecuredApp concrete class with factory method implemented	(2 points)
• LoginFrame class to build login frame	(3 points)
• RegisterFrame to build registration frame	(3 points)
• Create the factory method to return and display correct JFrame	(6 points)
• (Test) Run and test program to ensure factory method works	(2 points)