

CSE 5322: SOFTWARE DESIGN PATTERNS

Homework 1 15%

February 6, 2023

Due: 03/07/2023 11:59PM

1 What To Do

This individual homework assignment requires the student to design a simple application that applies the controller pattern, the expert pattern, the iterator pattern and the composite pattern. When the application starts, it displays a window containing two buttons and a drawing area. The buttons are labeled “Circle,” “Triangle” and “Box,” respectively. When one of the buttons is clicked, and then when the mouse is clicked in the drawing area, the corresponding shape is painted at the location clicked. As an illustration, Figure 1 shows a sample window with a circle and a box created. Note that the Triangle button is still highlighted because it was the last button that was clicked. The clicked button is highlighted automatically by the operating system.

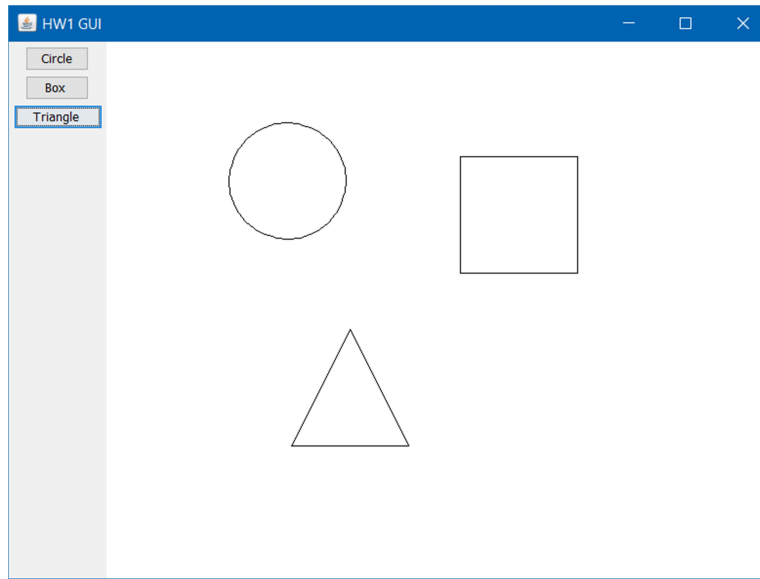


Figure 1: Sample window with a circle and box created

Useful hints:

1. Your design may need to remember which button is clicked so that when the mouse is clicked later in the drawing area your software will know which of the shapes is to be painted.
2. When the mouse is clicked in the drawing area, the clicked location must be captured and used to draw the shape at the location captured. To accomplish this, your scenario or high-level algorithm should (1) obtaining the location clicked, (2) determining which shape to create, (3) creating the shape with the location clicked, (4) add the shape object to the composite, and (5) repainting all of the shapes in the composite as described in the next bullet item.

Background information: In Java, the location clicked is automatically captured by a system-generated object called `MouseEvent`. Since it is system generated, it is also referred to as a system event — remember who should handle a system event. The `MouseEvent` object is automatically dispatched by the OS to a `MouseListener` subclass, which the application developer must implement to process the mouse event. The position clicked can be obtained by either calling the `getX()` and `getY()` methods or the `getPoint()` method of the `MouseEvent` object.

3. To paint the shape, **the drawing area is cleared first**, and then **each of the shapes stored in the composite is called to paint itself at its location**. This is accomplished by using the composite pattern.

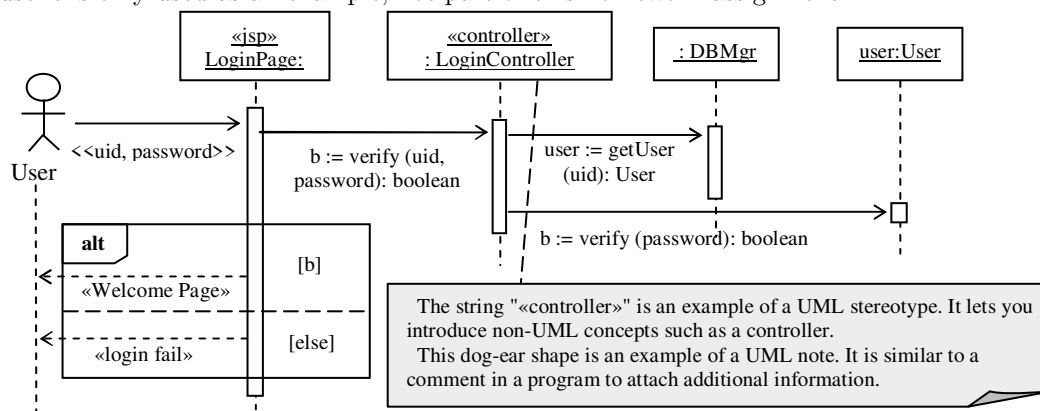
2 What to Do and Submit

The student is required to do and submit the following (equal weight):

1. Produce and submit a domain model class diagram (DMCD) for the simple application. Hint: The DMCD for this homework assignment **contains only a couple of classes**, that is, what the application is supposed to draw, **not the graphical user interface classes such as windows, drawing area and buttons**.

Note: the classes of the domain model should not show the methods, functions or operations. These will be referred to as methods in the rest of this document. Assigning methods to classes should take place during the design phase, that is, object interaction modeling (OIM) and deriving design class diagram (DCD). See the next work item for more information.

2. Produce and submit an expanded use case for the *Draw Shape* use case. This use case begins with the actor/user clicking one of the buttons. This causes the button clicked being highlighted. The actor then clicks anywhere in the drawing area. The system paints the desired shape at the position clicked. This use case ends with the actor/user seeing the corresponding shape painted at the position clicked in the drawing area.
3. Produce and submit a scenario for each nontrivial step of the expanded use case to describe how objects interact with each other to process the actor request. During this step, bear in mind to apply the controller, expert, iterator and composite patterns.
4. Convert each scenario to a scenario table and submit the resulting scenario tables.
5. Convert each scenario table to an informal sequence diagram and submit the resulting sequence diagrams.
6. Convert each informal sequence diagram to a design sequence diagram and submit the resulting design sequence diagrams. The design sequence diagram must indicate controller, expert, iterator and composite patterns that are applied by using either UML notes or UML stereotype. You may also use UML notes and stereotype to provide additional information/detail if needed. For your reference, the figure below illustrates examples of a UML stereotype and a UML note. Do not use or submit the following figure because it is only used as an example, not part of this homework assignment.



7. Produce and submit a design class diagram (DCD) derived from the design sequence diagrams and the domain model produced above. Again, use UML stereotype or UML note to show the patterns applied. You may also use UML notes and stereotype to provide additional information/detail if needed in the DCD.

8. Implement your design in Java and only Java — you cannot use any third party software except an IDE for coding and compilation. Start your software, draw one box and save the screen shot. Then draw two triangles and save the screen shot. Finally, draw three circles and save the screen shot.

Your implementation must include the graphical user interface such as the one shown in Figure 1. You may use swing or AWT.

3 How To Submit

Submit your analysis and design in one pdf or doc(x) file. Name your file as follows:

LastName_Firstname_CSE5322_S23_HW1.pdf, or
LastName_Firstname_CSE5322_S23_HW1.doc(x)

If you also implement the design, then compress your implementation files into one file and name it as follows:

LastName_Firstname_CSE5322_S23_HW1_impl.rar, or
LastName_Firstname_CSE5322_S23_HW1_impl.zip

Additional submission instructions may be provide by the GTA before the deadline.