Santa Clara University

Department of Computer Engineering

Client/Server and the Internet - (COEN 235)

Project-1 (10 pts)

1. Project Overview:

In this project, you will develop a *Clock* Bean and use the NetBeans IDE to demonstrate the *Clock* Bean in action. The *Clock* Bean can display time in either analog or digital forms. The *Clock* Bean inherits from the *Canvas* class, which derives from the *Component* class.

When the *Clock* is in analog mode, the hour, minute, and second hands need to sweep an appropriate amount as seconds tick away. When it's in digital mode, the number displayed must be updated with each passing second. To update once per second, the Bean implements the *Runnable* interface, which gives it a *run*() method executing within a thread that triggers an update every second.

2. Functional Requirements:

In analog mode, the *Clock* is a circle, it supports a font for displaying the hour number, which can be one of a Varity of different typefaces and sizes. The background color for the Bean is used in the area surrounding the *Clock*, which the foreground color determines the color of the hour number text. The background color of the *Clock* itself is always white, while the hour and minute arms are black and the second arm is red.

Clock Bean:

Properties: The Clock Bean inherits most of its properties from the *Canvas* class. The Bean has two additional **properties** including: *Raised* (true/false) that gives 3D effect around the Bean's border and *Mode* (analog/digital).

Methods: all properties of the Clock Bean are readable and writable (pair of accessor methods for each). In addition the Clock Bean needs additional 3 public methods:

- *run*(): thread that set up periodic trigger for clock updates.
- *update*(): implements an offscreen buffer
- *paint*(): draw the graphical representation of the Bean.

Events: no event related methods are needed.

3. Programming Hints:

Given that the *Clock* is updated once/second, to avoid flickering when repainting the Clock, you will need to implement an offscreen buffer. The *update*() method is called whenever the Bean needs to be repainted, e.g., when *repaint*() is called. The default *update*() in the Component class erases the background and then calls the *paint*() method. With using offscreen buffer to eliminate flicker, we don't want the background erased. As a result, you will need to override the *update*() method and instead just calls only the *paint*() method.

The *paint*() method is responsible for drawing the Bean, which includes the creation and manipulation of the offscreen buffer:

- Create/erase the offscreen buffer
- Draw at will to the offscreen buffer
- Draw the offscreen buffer to the screen.

4. Programming Model:

I have put in the corresponding course directory the ClockBean directory that contains the Source tree for all files needed to develop the Clock Bean. This includes: skeleton source file for the *Clock.java* file, the *TimerBeanInfo.java*, the *.gif files, and the *manifest* file (*Clock.mf*).

The *ClockBeanInfo* class was developed for you so the Bean has graphical icons associated with it to be used by the NetBeans. The Clock Bean also needs a *manifest* file to build a JAR file.

Navigate the directory tree, copy the tree in your own directory, complete the implementation of the *Clcok.java* file, compile all java files, and finally build the JAR file as follows:

ROOT: /users/faculty/aezzat/coen235/project-1/ClockBean

JAR file: $ROOT/Source/Clock.jar

Source code: $ROOT/Source/HTPJB/Clock/*

To build the *Clock.jar* yourself in your own private disk space:

```
% cd $ROOT/Source
% jar cfm Clock.jar
            HTPJB/Clock/Clock.mf
            HTPJB/Clock/*.class
            HTPJB/Clock/*.gif <cr>
```

To display the contents of the *Clock.jar*:

```
% cd $ROOT/Source
% jar tf Clock.jar
```

5. Project Execution:

- Create a project with NetBeans IDE. Follow the sequence of steps described in:
  - The Java Tutorial. Lesson: Using the NetBeans GUI Builder.
    https://netbeans.org/kb/docs/java/quickstart.html
    https://netbeans.org/kb/docs/java/gui-functionality.html
    http://docs.oracle.com/javase/tutorial/javabeans/quick/index.html
    http://wiki.netbeans.org/NetBeansJavaBeansTutorial
- Try both analog and digital Clock modes.