Práctica 14
Patrones

(Rev 1.0)

1 Objetivos

Que el alumno se familiarice con los patrones de diseño e implemente algunos de ellos en C++.

2 Pre-laboratorio

2.1 Cuestionario

(Las preguntas de los cuestionarios vendrán en sus exámenes parciales del salón.)

1. ¿Para qué se usa el patrón Singleton?
2. ¿Para qué se usa el patrón Observer?
3. ¿Para qué se usa el patrón Bridge?

2.2 Actividades

1. Escriba una implementación (ejemplo en C++) del patrón Bridge.

3 Teoría

3.1

4 Actividades

Actividad 1: Patrón Singleton

En esta actividad Ud. se familiarizará con el patrón Singleton a través de un modificación del ejemplo que recibió.

1. A partir de los ejemplos singleton.cpp y de reloj.cpp, escriba una clase singleton para tener una y únicamente una instancia de la clase Reloj.

2. Escriba un método .Comparar() que compare la hora de la clase singleton contra la hora recibida en el método. Como no puede haber más de un reloj, Ud. deberá seguir la siguiente firma:

```
bool Comparar (int hrs, int min, int s);
```

3. En su driver program ponga la hora del reloj , imprima la hora y compare la hora contra una hora arbitraria.

Actividad 2: Patrón Observer

En esta actividad Ud. programará una clásica aplicación de reloj alarma utilizando una variante (a nivel de utilización) del patrón Observer. En su celular Ud. puede establecer varias alarmas con la hora y el día; en esta actividad imitará dicho comportamiento (aunque únicamente utilizará la hora) a partir de los ejemplos observer.cpp y de reloj.cpp

1. El sujeto a ser observado será una instancia de la clase Reloj. Esta instancia lanzará una notificación cada minuto, y cada observador comparará la hora del sujeto contra la hora establecida en cada alarma. Si las horas coinciden, entonces mostrará un mensaje.

2. Los observadores serán instancias de una clase compuesta Alarma. La clase tendrá un dato miembro Reloj y un método Ejecutar(). El constructor deberá recibir la hora de la alarma, y el método Ejecutar() imprimirá una cadena en la pantalla.

3. En su driver program cree algunas alarmas y simule el paso del tiempo para que vea el funcionamiento de este patrón.

Actividad 3: Patrón Bridge

En esta actividad Ud. implementará a dicho patrón separando la interfaz de un reloj de su implementación. Existen al menos 3 implementaciones: un int para cada elemento de la hora; un arreglo de ints para los elementos de la hora; los segundos transcurridos desde la media noche.

1. Escriba una implementación del patrón Bridge tomando en cuenta lo ya explicado.
2. En su driver program pregunte al usuario con cuál de las tres implementaciones quiere trabajar. Cambie la hora, imprímala, increméntela, etc. para que vea el funcionamiento del patrón.