Universidad del Cauca
Departamento de Sistemas
Ingeniería de Software II

14 de enero de 2013 Profesor Julio Ariel Hurtado Alegría

Segundo Parcial

1. Conocimiento de Patrones de Diseño

Considere un simulador orientado a eventos genérico, es decir un simulador esqueleto que simula un rango amplio de sistemas. El simulador contiene una cola de eventos y una variable que indica lal hora de simulación(contador). Cada evento tiene grabado el tiempo que indica el momento en que éste debe ocurrir. La cola contiene eventos, los cuales son almacenados en orden ascendente resepcto al tiempo en que el evento ocurre. El simulador orientado a eventos genérico ejecuta el siguiente ciclo infinito en el método simular de la clase SimuladorGenerico:

```
abstract class SimuladorGenerico
 {
  protected Estado nuevoEstado, viejoEstado;
  protected double tiempoSimulado;
  protected ColaEventos colaEventos;
  protected Evento eventoActual;
  public abstract Estado cambiar(Estado viejoEstado, Evento evento);
  public abstract void actuar(Estado nuevoEstado, Estado viejoEstado);
  public void simular()
   {
    while (true)
    {
    tiempoSimulado = colaEventos.top().getTiempo();
     do
     {
       eventoActual = colaEventos.pop();
       nuevoEstado = cambiar(viejoEstado, eventoActual);
       actuar(nuevoEstado, viejoEstado);
       viejoEstado = nuevoEstado;
           } while(tiempoSimulado <= colaEventos.top().getTiempo());
```

```
    }
  }
}
```

a) El diseño del simulador orientado a eventos genérico esta basado parcialmente sobre un patrón de diseño. ¿Cuál es el patrón? motive su respuesta. Si está inseguro, motive otras alternativas que Usted considere.

b) En la clase SimuladorGenerico descrita arriba, ¿Por qué algunos métodos son abstractos?. Considere que Usted tiene que escribir un simulador para un sistema específico, por ejemplo la clase SimuladorBancoSimple (Un único Cajero, una única Fila (de clientes), eventos de llegada de un cliente y eventos de salida de un cliente de la atención del cajero). Dibuje una jerarquía de clases para el simulador específico, considere los posibles eventos y la lógica específica del simulador del banco.

c) ¿Podría usarse los patrones Fábrica Abstracta o el Método de Fábrica con el fin de generar eventos específicos al sistema simulado? Motive su respuesta.

d) ¿Cómo podría utilizar diferentes simuladores específicos usando el patrón de diseño Estrategia? Considere cambiar el código dado. Dibuje el diagrama de clases del sistema mejorado.

e) Hemos adquirido un sistema de gestión de bases de datos con una licencia para soportar 5 conexiones simultáneas. Con el fin de permitir muchas más conexiones simultáneas, aprovechando los tiempos muertos, se ha propuesto el desarrollo de una clase PoolConnections. ¿Qué patrón usaría para lograr esta solución? ¿Qué modificaciones se realizaron? Motive sus respuestas.

2. Caso de Aplicación de Patrones de Diseño

Considere el desarrollo de una aplicación para monitorear el desempeño de una aplicación multi-hilo, es decir una aplicación en la que se ejecutan varios procesos livianos al mismo tiempo, los cuales tienen cada uno una prioridad. Suponga que la programación del hilo se ha hecho en la clase Hilo, la cual almacena el estado y la prioridad, y tiene los métodos arrancar(prepara el hilo y llama al método correr), correr(donde normalmente se programa el código deseado por el hilo por las subclases), bloquear(esperando un recurso), desbloquear(cuando el recurso se libera o

cuando el hilo es forzado), dormir(se lo deja esperando por un tiempo sin hacer nada) y finalizar (llamado antes de morir para liberar recursos), los cuales producen cambio en el estado del hilo. El monitoreo se hace a través de un visualizador de los hilos(VisualizadorHilos), con el fin de saber visualmente el estado de los hilos. Es decir, si cada uno de éstos está en preparación(starting-amarillo), está activo(running-verde), está durmiendo(sleeping-azul) o está bloqueado(Waiting-rojo). Esto con el fin de evaluar el desempeño del sistema (consumo de tiempo) y poder realizar ciertas acciones como matar un hilo (kill), desbloquearlo o cambiarle la prioridad (por ejemplo bajarle de prioridad a un hilo voraz con alta prioridad). El visualizador también debe poder evaluar y ejecutar éstas mismas acciones sobre varios hilos, por lo cual los hilos son organizados en forma de grupos(GrupoHilo). Por defecto los hilos o grupos hilo, son creados en la aplicación principal(Principal) donde pertenece el hilo principal del programa (por defecto el sistema crea el grupoPrincipal), a menos que se le indique otro grupo al cuál pertenece.

a) Muestre mediante un diagrama de clases como el patrón composité puede ayudarle a organizar los hilos para lograr un manejo homegéneo de los mismos. Valor 0.5

b) Complete el diagrama de clases anterior considerando la clase VisualizadorHilos haciendo uso del patrón MVC simplificado(Observador e Intermediario), usando un controlador intermedio entre el visualizador y los objetos hilo. Valor 1.0.

c) Haga un diagrama de secuencia que ilustre el comportamiento del sistema de monitoreo. Valor 0.5

d) Programe en su lenguaje de programación orientado a objetos favorito las clases Hilo, GrupoHilo, Visualizador y Controlador. Programe únicamente la lógica básica suficiente para demostrar la implementabilidad del sistema que Usted propone. Valor 1.0

Puro dibujo es una abstracción.
El dibujo y el color no son contrarios, todo en la naturaleza es color.
Paul Cezanne