UNIVERSIDAD DE COSTA RICA
Escuela de Ciencias de la Computación e Informática **I semestre 2024**
CI 0136 Diseño de Software
Prof. Mauricio Arroyo (mauricio.arroyo@ucr.ac.cr)

Laboratorio 7: Behavioral Patterns
(8 puntos relativos)

Objetivo:

Que el estudiante tenga un acercamiento con algunos behavioral patterns; y los tenga como parte de sus herramientas cuando realice el diseño de un sistema.

Indicaciones generales:

- Para este laboratorio usted trabajará con su compañer@ de laboratorio. Es importante que ambos participen.
- Siga los pasos descritos en la siguiente sección.
- Al finalizar el laboratorio cada grupo debe subir un archivo zip con cada patrón funcionando.

Laboratorio

Strategy

1. Abra el código TaxCalculator.zip
2. Usted ha sido contratado para diseñar e implementar un prototipo de un sistema que se encarga de determinar el costo de una factura, incluyendo el impuesto de ventas.
3. Su sistema será utilizado en diferentes países, donde el impuesto de ventas varía en porcentaje.
   Estados Unidos: 5%
   UK: 7%
   Costa Rica: 13%
4. Abra el código del laboratorio y estúdielo.
5. Modifique el switch del cálculo del impuesto por el manejo del patrón Strategy.

Observer

1. Abra el código Subasta.zip
2. Después de empezar el desarrollo de un nuevo sistema para simular subastas. Su jefe se da cuenta que la manera en que se están manejando los licitadores y quienes deben ser notificados de nuevas ofertas es poco mantenible. Cada nuevo tipo de licitador implica modificar el código que ya estaba desarrollado.
3. Le ha pedido a usted cambiar ese código para utilizar el patrón de Observer.
4. Estudie el código, modifíquelo para que utilice el patrón de Observer.

5. Agregue un nuevo tipo de licitador con un comportamiento propuesto por ustedes.

Command Pattern

1. Abra el código "RemoteControl.zip"
2. Su empresa fue contratada para implementar un control remoto. Y quieren que los botones de este control remoto sean personalizables. Es decir que usted pueda asignar diferentes aspectos de su casa inteligente a un slot del control dependiendo de su gusto.
3. Para esto se le entregaron tres clases que controlan 3 dispositivos de una casa:
    a. La puerta del garaje
    b. Una luz (a la cual se le puede especificar el espacio al que pertenece)
    c. Un estéreo.
4. Para facilitar el desarrollo del laboratorio se provee también como parte del mismo la clase remoteControl, que contiene el array de comandos, y el código del manejo de dichos comandos, deben tomar en cuenta lo siguiente:
    a. El código para el manejo de los comandos se provee para agilizar el proceso del laboratorio, pero de ninguna manera esto hace que no tengan que entender que está pasando en ese código.
    b. Abran el código y analícenlo. En particular no pierda de vista que se está utilizando el patrón de NullObject para asignar comandos vacíos a cada slot del control remoto al iniciar el mismo.
5. Debe implementar 8 comandos, utilizando el patrón, uno para cada slot en uso del control remoto.
6. Modifique el método main, para realizar la asignación de los comandos a los slots utilizando el método setCommand del control remoto. Y llamados a los métodos de cada slot para probar que se ejecuta el método correcto.

Template Pattern

1. Abra el código "ScoringCalculator.zip"
2. Este código provee la funcionalidad básica para calcular el puntaje total en una competencia de tiro al blanco. Las reglas del juego son las siguientes:
    a. **Hombres:** Cien puntos por cada objetivo atinado. Cinco puntos menos por cada segundo de tiempo utilizado.
    b. **Mujeres:** Cien puntos por cada objetivo atinado. Diez puntos menos por cada segundo de tiempo utilizado (más que los hombres para que no me acusen de machista)
    c. **Niños:** Doscientos puntos por cada objetivo atinado. Dos puntos menos por cada segundo de tiempo utilizado.

    Los scores negativos son reemplazados por cero.
3. Implemente lo necesario para utilizar el template método GenerateScore, incluyendo la definición de las operaciones primitivas utilizadas en este método y que deben ser redefinidas en las clases derivadas. Defina las clases MenScoringAlgorithm, WomenScoringAlgorithm y ChildrenScoringAlgorithm, de manera que se pueda ejecutar el código en el método main, que prueba su implementación.
4. El resultado del método main debe ser:

```
Man 782
Woman 873
```

Child 909

State Pattern

1. Abra el código “RPGGame.zip”
2. Este código provee la funcionalidad básica del manejo de estados para el personaje de un juego, en nuestro caso un hombre lobo.
3. Sin embargo, nuestro diseño actual es inflexible, y sabiendo que los dueños de la empresa van a querer más acción en el juego, y por lo tanto nuevos movimientos de los personajes, hemos decidido cambiar el diseño para utilizar el patrón State.
4. Realice los cambios necesarios para aplicar este patrón, cambiando los if’s de los cambios de estado de la clase personaje. Recuerde que las transiciones de los estados pueden tener un parámetro apuntando a su contexto (en este caso la clase Personaje) para obtener información de él.