Gregorio Vidoy Fajardo

Ejercicio 4 P1

Tabla de contenidos:

DESCRIPCIÓN DEL PROBLEMA:

Utilizando el patrón arquitectónico “Interceptor” aplicado a una parte del problema de control SCACV ("sistema de control automático para la conducción de un vehículo"), desarrollar un diagrama de clases y un proyecto de Eclipse/Java para calcular la velocidad inicial del vehículo a partir de un dato de entrada, p.e.: revoluciones del eje, instalando posteriormente un manejador de eventos que "reaccione" cuando se pulsen cualquiera de los 2 botones: ``Encender" (el motor del vehículo) y "Acelerar" la velocidad de crucero.

Para que el ejercicio sea considerado correcto hay que realizarlo de acuerdo con los siguientes requerimientos:

-Programar una clase anónima (WindowAdapter()) con Swing (Java) para terminar bien la ejecución de la clase Interfaz correspondiente:

```
this.addWindowListener (new WindowAdapter(){

  public void windowClosing(WindowEvent e){

    System.exit(0);

  }

});
```

Hay que programar los botones “Encender”, “Acelerar” y la etiqueta “APAGADO” / “ACELERANDO” dentro de un objeto panel de botones, cuyo esqueleto en Swing sería algo como lo siguiente:

```
import java.awt.*;import javax.swing.BoxLayout;

import javax.swing.JPanel;import javax.swing.border.*;

public class PanelBotones extends JPanel {

  private javax.swing.JButton BotonAcelerar, BotonEncender;

  private javax.swing.JLabel EtiqMostrarEstado;

  public PanelBotones(){ ...};//constructor

  synchronized private void

     BotonAcelerarActionPerformed(java.awt.event.ActionEvent evt){...};
```

```
 synchronized private void

    BotonEncenderActionPerformed(java.awt.event.ActionEvent evt){...};

}
```

Funcionamiento de los botones:

-Inicialmente la etiqueta del panel principal mostrará el texto "APAGADO" (ver figura a) y las etiquetas de los botones, el nombre de cada uno.

-El botón "Encender" será de selección de tipo conmutador JToggleButton, cambiando de color y de texto ("Encender"/"Apagar") cuando se pulsa.

-La pulsación del botón "Acelerando" cambia el texto de la etiqueta del panel principal a "ACELERANDO" (ver figura b), pero sólo si el motor está encendido; si no, no hace caso a la pulsación del usuario.

-Si ahora se pulsa el botón que muestra ahora la etiqueta "Apagar", la etiqueta del panel principal volverá a mostrar el texto inicial "APAGADO".

RELEASE NOTE:

-Compilado con Java 1.8
-Clase Main: Demostracion

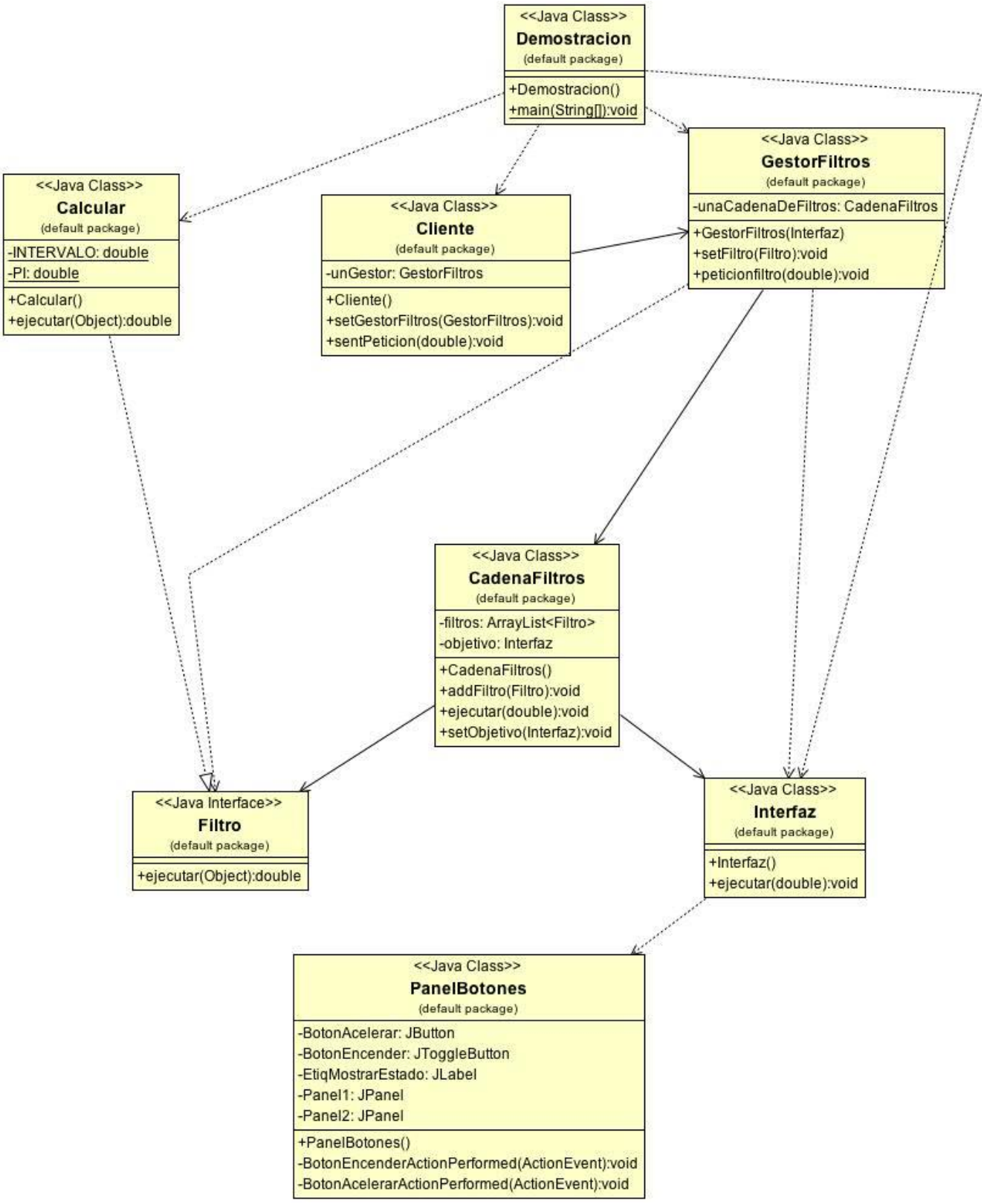
SOLUCIÓN AL PROBLEMA:

Para la solución del problema he hecho uso del patrón interceptor, para ello encapsulo una operación en un filtro, definido como una Interface que implementa calcular, el cual se inserta en una cadena de filtros, para además así poder restaurar un estado anterior.

EXPLICACIÓN DEL PROGRAMA:

El programa instancia un gestor de filtros, el cual instancia una Interfaz, inserta el filtro calcular, e instanciamos un cliente que será el manejador y realiza una petición.
Por otro lado, la interfaz gráfica que cuenta con dos botones jButton y una etiqueta jLabel, que muestran el estado inicial apagado. Cuando pulsamos el botón acelerar no hace nada si antes no estaba pulsado encender, si pulsamos encender la etiqueta cambia a encendido, y si ahora aceleramos cambia a acelerando.

UML:


<<Java Class>>
Demostracion
(default package)
+Demostracion()
+main(String[]):void
<<Java Class>>
GestorFiltros
(default package)
-unaCadenaDeFiltros: CadenaFiltros
+GestorFiltros(Interfaz)
+setFiltro(Filtro):void
+peticionfiltro(double):void
<<Java Class>>
Calcular
(default package)
-INTERVALO: double
-PI: double
+Calcular()
+ejecutar(Object):double
<<Java Class>>
Cliente
(default package)
-unGestor: GestorFiltros
+Cliente()
+setGestorFiltros(GestorFiltros):void
+sentPeticion(double):void
<<Java Class>>
CadenaFiltros
(default package)
-filtros: ArrayList<Filtro>
-objetivo: Interfaz
+CadenaFiltros()
+addFiltro(Filtro):void
+ejecutar(double):void
+setObjetivo(Interfaz):void
<<Java Interface>>
Filtro
(default package)
+ejecutar(Object):double
<<Java Class>>
Interfaz
(default package)
+Interfaz()
+ejecutar(double):void
<<Java Class>>
PanelBotones
(default package)
-BotonAcelerar: JButton
-BotonEncender: JToggleButton
-EtiqMostrarEstado: JLabel
-Panel1: JPanel
-Panel2: JPanel
+PanelBotones()
-BotonEncenderActionPerformed(ActionEvent):void
-BotonAcelerarActionPerformed(ActionEvent):void