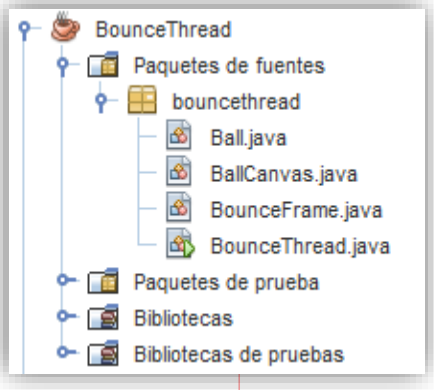
Laboratorio 04
Ejemplo con Hilos y Canvas en Java



Rebote con hilos (BounceThread)

BounceThread.java

```java
package bouncethread;

import javax.swing.JFrame;

public class BounceThread { //Rebote hilado
    public static void main(String[] args) {
        JFrame frame = new BounceFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.show(); // Mostrando la estructura gráfica
    }
```

BounceFrame.java

```java
package bouncethread;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class BounceFrame extends JFrame{ // Estructura con el canvas y botones
    private BallCanvas canvas; // Referencia a la clase BallCanvas
    public static final int WIDTH = 450;
    public static final int HEIGHT = 350;
    /*
    * Construye la estructura con el area canvas para mostrar el balon y los
    * botones Start y Close. La clase Canvas provee un area rectangular con
    * capacidad para que la aplicacion pueda dibujar graficos e imagenes y
    * atrapar eventos del usuario.
    */
    public BounceFrame(){ // Constructor
        setSize(WIDTH, HEIGHT);
        setTitle("BounceThread");
        Container contentPane = getContentPane(); // Se define el contenedor
        canvas = new BallCanvas(); // instanciamos BallCanvas

        // Incorporamos canvas al contentPane, region CENTER
        contentPane.add(canvas, BorderLayout.CENTER);
        JPanel buttonPanel = new JPanel(); // Panel para botones

        // Incorporamos el boton start al buttonPanel
        addButton(buttonPanel, "Start",new ActionListener(){
        public void actionPerformed(ActionEvent evt){addBall();}});

        // Incorporamos el boton close al buttonPanel
        addButton(buttonPanel, "Close",new ActionListener(){
            public void actionPerformed(ActionEvent evt){System.exit(0);}
        });

        // Incorporo buttonPanel al contentPane, region SOUTH
        contentPane.add(buttonPanel, BorderLayout.SOUTH);
    }
```


```
39          // El método addButton propio
40          public void addButton(Container c, String title, ActionListener listener){
41              JButton button = new JButton(title);
42              c.add(button);
43              button.addActionListener(listener);
44          }
45          // Incorporamos el balon al canvas y arrancamos el hilo de rebotes
46          public void addBall(){
47              Ball b = new Ball(canvas);
48              canvas.add(b);
49              BallThread thread = new BallThread(b);
50              thread.start();
51          }
52      } // Final class BounceFrame
53
54      // El hilo para jugar con el balon...
55      class BallThread extends Thread{ // class BallThread
56          private Ball b;
57          public BallThread(Ball aBall) {
58              b = aBall;
59          }
60          public void run(){
61              try{
62                  for (int i = 1; i <= 10000; i++){
63                      b.move(); sleep(5);
64                  }
65              } catch (InterruptedException exception){}
66          }
67      }
```

Ball.java

```
1   package bouncethread;
2
3   import java.awt.Graphics2D;
4   import java.awt.Component;
5   import java.awt.geom.*;
6
7   class Ball{ // Clase balon
8       private Component canvas;
9       private static final int XSIZE = 15; private static final int YSIZE = 15;
10      private int x = 0; private int y = 0; private int dx = 2; private int dy= 2;
11      public Ball(Component c) { // Construimos un balon
12          canvas = c;
13      }
14      public void draw(Graphics2D g2){ // Dibujamos balon, posición corriente
15          g2.fill(new Ellipse2D.Double(x, y, XSIZE, YSIZE));
16      }
17      public void move(){ // Movemos balon
18          x += dx; y += dy;
19          if (x < 0){
20              x = 0; dx = -dx;
21          }
22          if (x + XSIZE >= canvas.getWidth()){
23              x = canvas.getWidth() - XSIZE; dx = -dx;
24          }
25          if (y < 0){
26              y = 0; dy = -dy;
27          }
28          if (y + YSIZE >= canvas.getHeight()){
29              y = canvas.getHeight() - YSIZE; dy = -dy;
30          }
```

```
31              canvas.repaint();
32          }
33      }
```

BallCanvas.java

```
1   package bouncethread;
2   // El canvas para dibujar el balon
3   import javax.swing.*;
4   import java.util.*;
5   import java.awt.*;
6
7   class BallCanvas extends JPanel{ // class BallCanvas
8       /*
9       * Implementamos ArrayList, un array redimensionable vacio que pueda
10      * contener objetos
11      */
12      private ArrayList balls = new ArrayList();
13      // Le adicionamos un balon
14      public void add(Ball b){
15          balls.add(b);
16      }
17      public void paintComponent(Graphics g){ // Dibujamos el balon
18          super.paintComponent(g);
19          Graphics2D g2 = (Graphics2D)g;
20          for (int i = 0; i < balls.size(); i++){
21              Ball b = (Ball)balls.get(i); b.draw(g2);
22          }
23      }
24  }
```