

LAB 04 – MULTITHREAD



Bài 1: Viết chương trình sử dụng SingleThread để tạo ra một quả banh chạy trên màn hình.

Bước 1: Tạo ra lớp Ball.java

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

public class Ball {
    private JPanel box;
    private static final int XSIZE=10;
    private static final int YSIZE=10;
    private int x=0;
    private int y=0;
    private int dx=2;
    private int dy=2;

    public Ball(JPanel p) {
        box=p;
    }

    public void draw(){
        Graphics g=box.getGraphics();
        g.fillOval(x, y, XSIZE, YSIZE);
        g.dispose();
    }

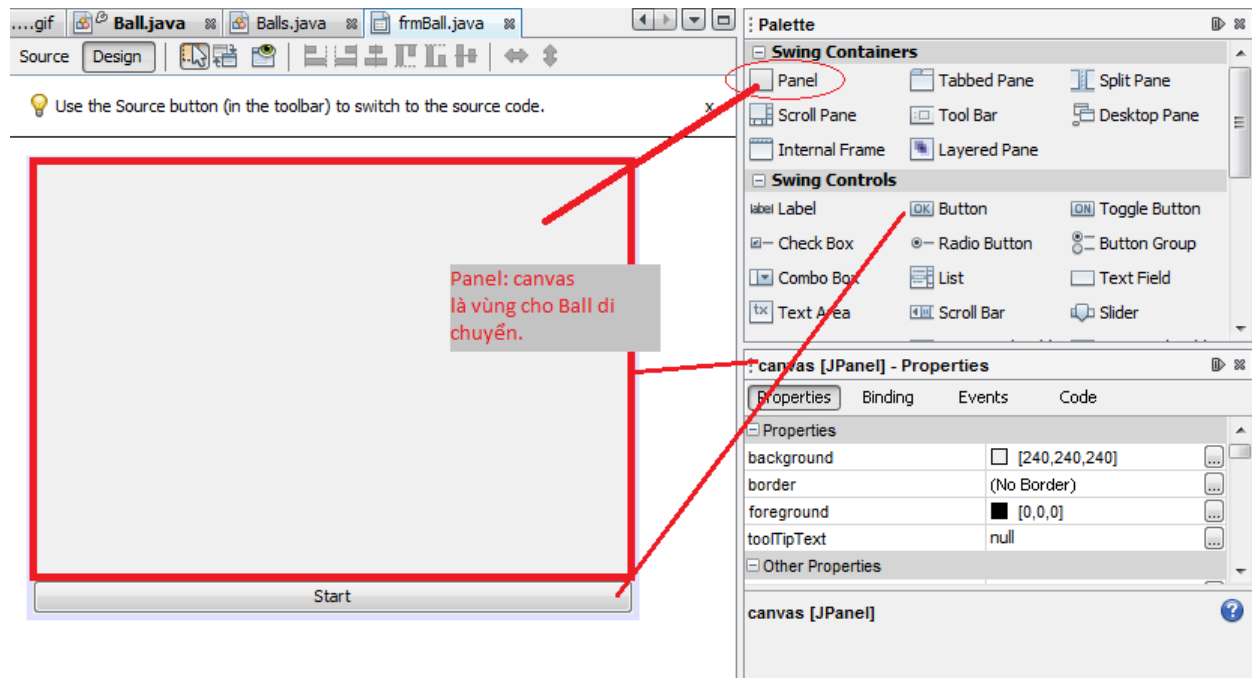
    public void move(){
        //xoa hình cũ bằng cách vẽ đè lên
        Graphics g=box.getGraphics();
        g.setXORMode(Color.CYAN);
        g.fillOval(x, y, XSIZE, YSIZE);
        x+=dx;
        y+=dy;
        Dimension d=box.getSize();
        //kiemtra cos dung cac canh
        if(x<0){
```

```
        x=0;
        dx=-dx;
    }
    if(x+XSIZE>=d.getWidth()){
        x=d.width-XSIZE;
        dx=-dx;
    }
    if(y<0){
        y=0;
        dy=-dy;
    }

    if(y+YSIZE>=d.getHeight()){
        y=d.height-YSIZE;
        dy=-dy;
    }
    g.fillOval(x, y, dx, XSIZE);
    g.dispose();
}

public void bounce(){
    draw();
    for(int i=0;i<1000;i++){
        move();
        try{
            Thread.sleep(5);
        }catch(InterruptedException ex){
            JOptionPane.showMessageDialog(null, ex.toString(), "Thông báo
            lỗi", JOptionPane.ERROR_MESSAGE);
        }
    }
}
}
```

Bước 2: Tạo Form có giao diện như sau:



Bước 3: Xử lý sự kiện cho button Start:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Ball b1=new Ball(canvas);  
    b1.bounce();// TODO add your handling code here:  
}
```

Bước 4: Chạy thử ứng dụng bằng cách nhấp nhiều lần vào button Start.

Bài 2: Viết chương trình sử dụng Multithread để tạo ra nhiều quả banh chạy trên màn hình.

Bước 1: Tạo ra lớp Balls.java

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

public class Balls extends Thread{
    private JPanel box;
    private static final int XSIZE=10;
    private static final int YSIZE=10;
    private int x=0;
    private int y=0;
    private int dx=2;
    private int dy=2;

    public Balls(JPanel p) {
        box=p;
    }

    public void draw(){
        Graphics g=box.getGraphics();
        g.fillOval(x, y, XSIZE, YSIZE);
        g.dispose();
    }

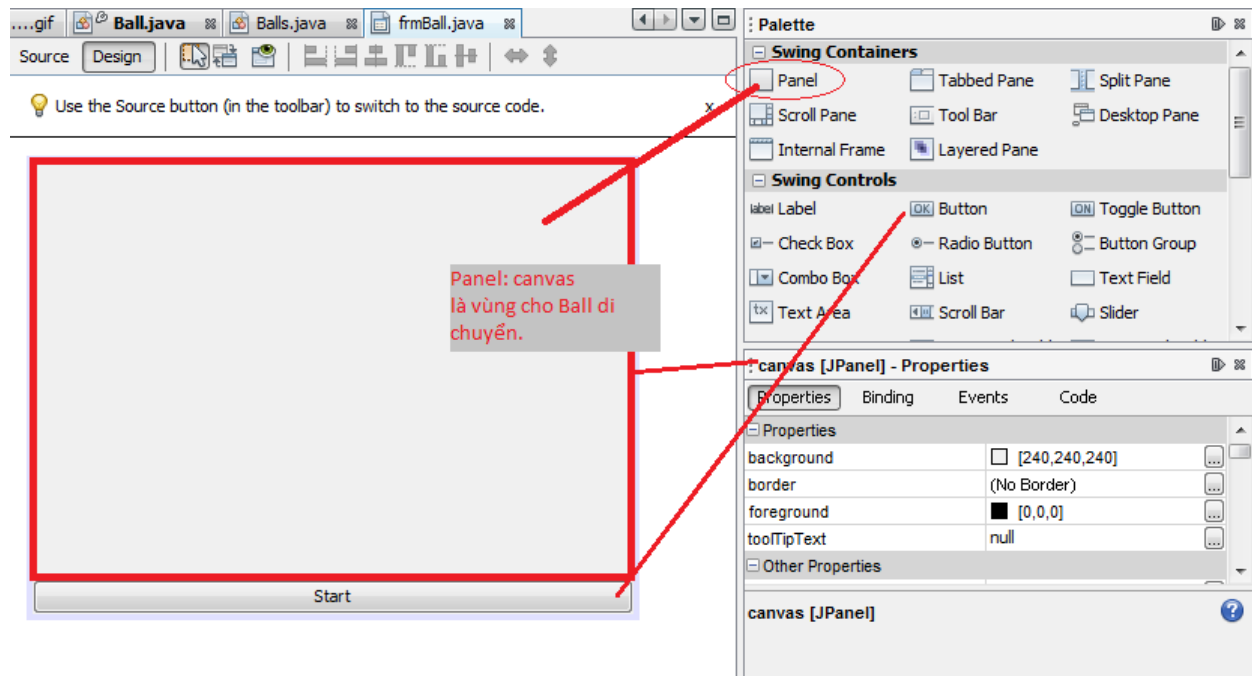
    public void move(){
        //xoa hinh cu bang cach ve de len
        Graphics g=box.getGraphics();
        g.setXORMode(Color.GREEN);
        g.fillOval(x, y, XSIZE, YSIZE);
        x+=dx;
        y+=dy;
        Dimension d=box.getSize();
        //kiemtra cos dung cac canh
        if(x<0){
            x=0;
            dx=-dx;
        }
        if(x+XSIZE>=d.getWidth()){
            x=d.width-XSIZE;
            dx=-dx;
        }
        if(y<0){
```

```
        y=0;
        dy=-dy;
    }

    if(y+YSIZE>=d.getHeight()){
        y=d.height-YSIZE;
        dy=-dy;
    }
    g.fillOval(x, y, dx, XSIZE);
    g.dispose();
}

public void run(){
    draw();
    for(int i=0;i<5000;i++){
        move();
        try{
            sleep(1);
        }catch (InterruptedException ex){
            JOptionPane.showMessageDialog(null, ex.toString(),"Thông báo
            lỗi",JOptionPane.ERROR_MESSAGE);
        }
    }
}
}
```

Bước 2: Tạo Form có giao diện như sau:



Bước 3: Xử lý sự kiện cho button Start:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Balls b=new Balls(canvas);
    b.start();// TODO add your handling code here:
}
```

Bước 4: Chạy thử ứng dụng bằng cách nhấp nhiều lần vào button Start.

Bài 3. Viết chương trình tạo ra một đối tượng `FileTWrite` kế thừa đối tượng `Thread` (hoặc cài đặt giao diện `Runnable`), cho phép viết một dãy số ngẫu nhiên vào tập tin. Đối tượng `FileTWriter` có thuộc tính tên tập tin cần viết. Viết hàm `main` tạo ra 3 đối tượng viết 3 tập tin chạy ở 3 tiến trình đồng thời riêng biệt.

Bài 4. Viết chương trình tạo ra một đối tượng `FileTReader` kế thừa đối tượng `Thread` (hoặc cài đặt `Runnable`), cho phép đọc nội dung một tập tin và hiển thị lên màn hình. Đối tượng `FileTReader` có thuộc tính tên tập tin cần mở. Viết hàm `main` tạo ra 3 đối tượng đọc 3 tập tin chạy ở 3 tiến trình đồng thời riêng biệt.

Bài 5. Viết chương trình tạo ra đối tượng đọc tập tin, viết tập tin chạy ở từng tiến trình riêng (có xử lý đồng bộ hóa dữ liệu). Viết hàm `main` tạo ra đối tượng viết và đọc dữ liệu với tập tin giống nhau. Kiểm tra việc đồng bộ hóa dữ liệu.