

2020-2

플랫폼기반프로그래밍

Week11

Graphics & Thread

담당교수 : 정은미



부산대학교
PUSAN NATIONAL UNIVERSITY

실습 목표

1. Interface와 Lambda Expression를 이용하여 프로그램을 구현해 본다.
2. 스레드와 Graphics를 활용하여 공을 움직이는 프로그램을 작성할 수 있다.

참고.

Lambda Expression

- Lambda Expression이란 식별자 없이 실행 가능한 함수이다.
- 함수적 프로그래밍을 위해 자바8부터 Lambda Expression을 지원 한다.
- Lambda Expression은 익명함수(anonymousfunction)을 생성하기 위한 식으로 객체지향언어보다는 함수 지향 언어에 가깝다.
- Lambda Expression의 형태는 매개변수를 가진 코드 블록이지만, 런 타임 시에는 익명구현 객체 (Abstract Method를 한 개 포함한)를 생성한다.
- Lambda Expression을 사용하면 기존의 불필요한 코드를 줄이고 가독성을 향상시킬 수 있다.

Functional Interface

- Abstract Method가 하나인 인터페이스이다.
- Lambda Expression사용이 가능하다.
- @FunctionalInterface annotation을 붙이지 않아도 되지만, 붙이는 것을 권장함.

예제

```
@FunctionalInterface
interface Func{
    public int calc(int a, int b);
}

public class Lambda {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Func add=(a,b)->a+b;
        Func sub=(a,b)->a-b;

        System.out.println(add.calc(3, 2));
        System.out.println(sub.calc(2, 1));
    }
}
```

Problems @ Javadoc Declaration Console

<terminated> Lambda [Java Application] C:\Program Files\Java\

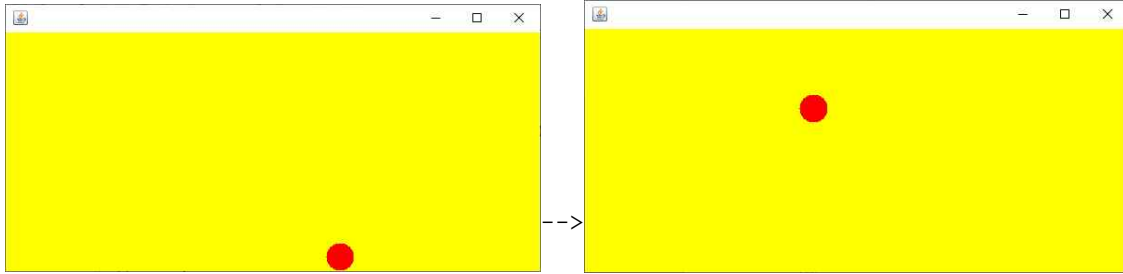
5

1

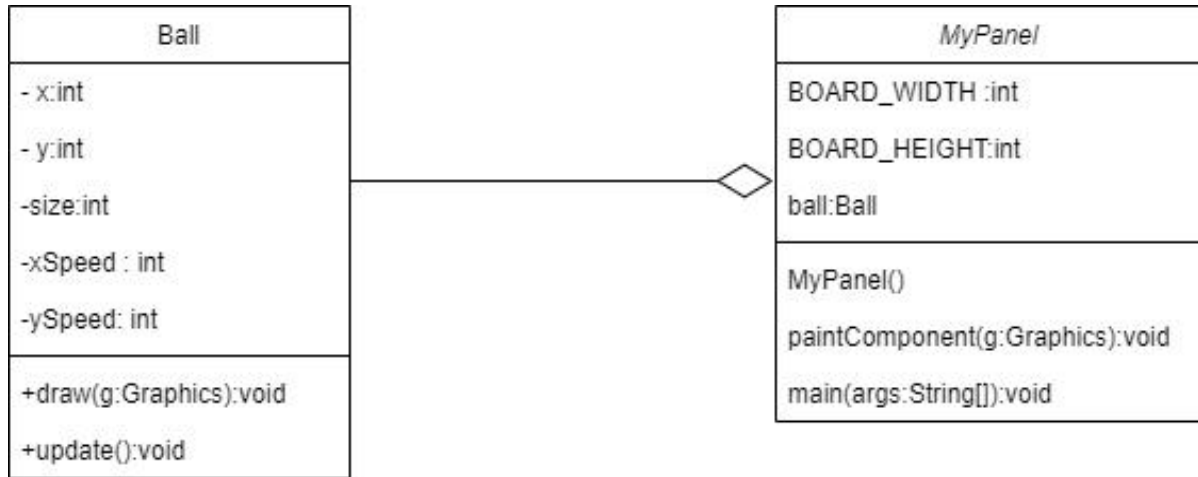
Lab. 공 움직이기

Graphics와 Thread를 사용하여, 화면에서 공을 움직이는 프로그램을 작성하시오.

- 스레드의 작업을 지정할 때 람다식을 이용해 보자



다음과 같은 UML 클래스 다이어그램을 참조한다.



요구 조건(반드시 준수)

- 명시한 클래스를 반드시 구현하여 사용
- Lambda expression을 사용하여 Runnable객체를 생성하는 코드는 다음과 같다.

```
Runnable task=()->{
    while(true) {
        //ball객체의 update()호출
        //다시 그리기
        try {
            Thread.sleep(50);
        }catch(InterruptedException e) {
        }
    }
};
```

- 아래 코드를 완성하세요.

```

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

//공을 Ball 클래스로 모델링한다.

class Ball {

//공의 속성은 위치, 크기, 속도이다.
    private int x = 100;
    private int y = 100;
    private int size = 30;
    private int xSpeed = 10;
    private int ySpeed = 10;

//공을 화면에 그려주는 메소드이다.

    public void draw(Graphics g) {
        //여기에 코드를
        //그래픽 색상지정
        //속이 꽉찬 원 그리기 , (x,y)위치에 sizeXsize크기
    }

    public void update() {
//공의 새로운 위치를 계산한다. 공이 벽에 부딪치면 반사되도록 한다.
//공은 xSpeed, ySpeed만큼 위치가 이동한다.
//xSpeed, ySpeed, x,y,BOARD_WIDTH, BOARD_HEIGHT값을 사용하여 작성
    }
}

public class MyPanel extends JPanel {
    static final int BOARD_WIDTH = 600;
    static final int BOARD_HEIGHT = 300;
    private Ball ball = new Ball();

    public MyPanel() {
        this.setBackground(Color.YELLOW);
        //람다식을 이용하여서 Runnable 객체를 생성한다.
        //무한루프를 돌면서 공의 위치를 변경하고 화면에 다시 그린다. 50 밀리초동안 쉰다.

        // 스레드 객체를 생성하고 스레드를 실행시킨다.
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
    }
}

```

```

        ball.draw(g);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.setSize(MyPanel.BOARD_WIDTH, MyPanel.BOARD_HEIGHT);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
        frame.add(new MyPanel());
    }
}

```

- 도전 문제** : 1. 화면에 공을 여러 만들려면 어떻게 하면 좋을까? 공의 색상도 랜덤하게 나오게 하려면 어떻게 하면 좋을까?(가산점 : 2점)
2. Start버튼과 Stop버튼을 만들어서 Start버튼이 눌러진 경우에만 공이 움직이도록 해 보자. Stop버튼이 눌러지면 공이 정지하여야 한다.(가산점 : 3점)

도전 과제는 14주까지 제출 시 해당 가산점 부여.

제출방법 : 실습게시판에 업로드