

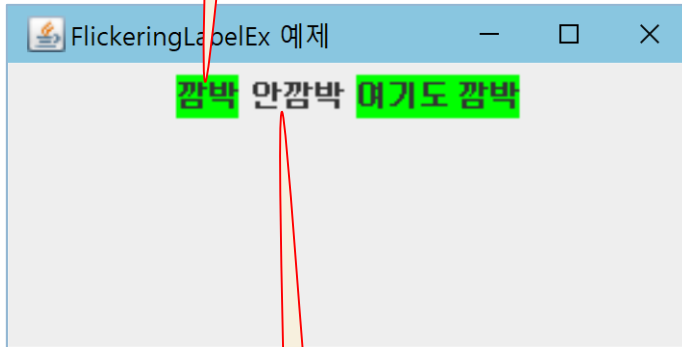
예제 13-3 : 깜박이는 문자열을 가진 레이블 만들기

1

JLabel을 상속받아 문자열을 깜박이는 FlickeringLabel 컴포넌트를 작성하라.

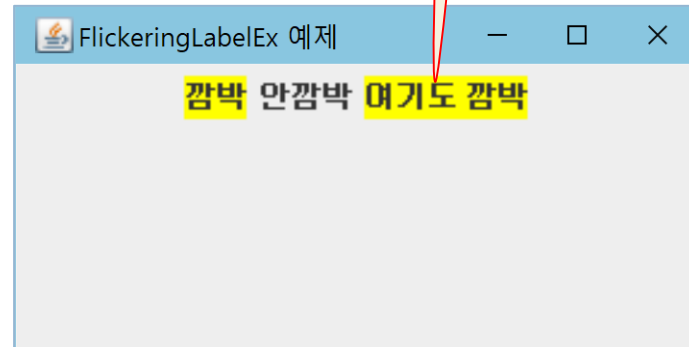
➔ 두 개의 레이블을 200밀리와 350밀리 주기로 깜박이게 변경하고 다른 바탕 색을 설정하시오

500 밀리 주기로 초록색과 노란색으로
번갈아 깜박이는 레이블



깜박이지 않는 레이블

300 밀리 주기로 초록색과 노란색으로
번갈아 깜박이는 레이블



예제 13-3 : 정답 코드

2

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

```
class FlickeringLabel extends JLabel implements Runnable {
    private long delay;
    public FlickeringLabel(String text, long delay) {
        super(text);
        this.delay = delay;
        setOpaque(true);
        Thread th = new Thread(this);
        th.start();
    }
    @Override
    public void run() {
        int n=0;
        while(true) {
            if(n == 0)
                setBackground(Color.YELLOW);
            else
                setBackground(Color.GREEN);
            if(n == 0) n = 1;
            else n = 0;
            try {
                Thread.sleep(delay);
            }
            catch(InterruptedException e) {
                return;
            }
        }
    }
}
```

```
public class FlickeringLabelEx extends JFrame {
    public FlickeringLabelEx() {
        setTitle("FlickeringLabelEx 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        // 깜박이는 레이블 생성
        FlickeringLabel fLabel = new FlickeringLabel("깜박",500);

        // 깜박이지 않는 레이블 생성
        JLabel label = new JLabel("안깜박");

        // 깜박이는 레이블 생성
        FlickeringLabel fLabel2 = new FlickeringLabel("여기도 깜박",300);

        c.add(fLabel);
        c.add(label);
        c.add(fLabel2);

        setSize(300,150);
        setVisible(true);
    }

    public static void main(String[] args) {
        new FlickeringLabelEx();
    }
}
```

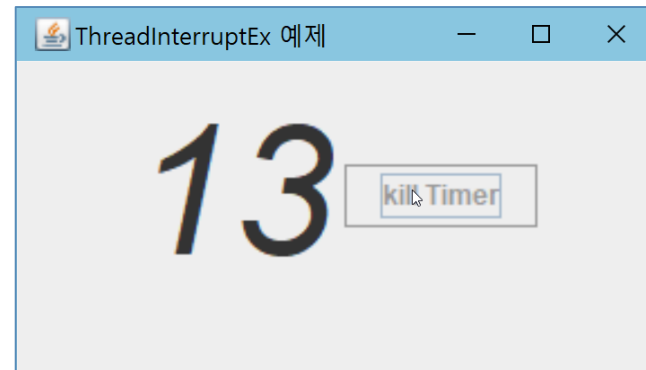
예제 13-5 : 타이머 스레드 강제 종료

3

아래 그림과 같이 작동하여 타이머 스레드를 강제 종료시키는 스윙 응용프로그램을 작성해보자.
➔ Kill Timer 버튼을 2분 내로 클릭하지 않는 경우 타이머를 멈추고 버튼을 비활성화 하도록 프로그램을 수정 하시오



타이머는 정상 작동한다.



Kill Timer 버튼을 클릭하면
타이머가 멈춘다.
버튼은 비활성화된다.

예제 13-5 : 정답 코드

4

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

class TimerRunnable implements Runnable {
    private JLabel timerLabel;

    public TimerRunnable(JLabel timerLabel) {
        this.timerLabel = timerLabel;
    }

    @Override
    public void run() {
        int n=0;
        while(true) {
            timerLabel.setText(Integer.toString(n));
            n++;
            try {
                Thread.sleep(1000); // 1초 동안 잠을 잔다.
            }
            catch (InterruptedException e) {
                return; // 예외가 발생하면 스레드 종료
            }
        }
    }
}
```

```
public class ThreadInterruptEx extends JFrame {
    private Thread th;
    public ThreadInterruptEx() {
        setTitle(" ThreadInterruptEx 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        JLabel timerLabel = new JLabel();
        timerLabel.setFont(new Font("Gothic", Font.ITALIC, 80));

        TimerRunnable runnable = new TimerRunnable(timerLabel);
        th = new Thread(runnable); // 스레드 생성
        c.add(timerLabel);

        // 버튼을 생성하고 Action 리스너 등록
        JButton btn = new JButton("kill Timer");
        btn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                th.interrupt(); // 타이머 스레드 강제 종료
                JButton btn = (JButton)e.getSource();
                btn.setEnabled(false); // 버튼 비활성화
            }
        });
        c.add(btn);
        setSize(300,170);
        setVisible(true);

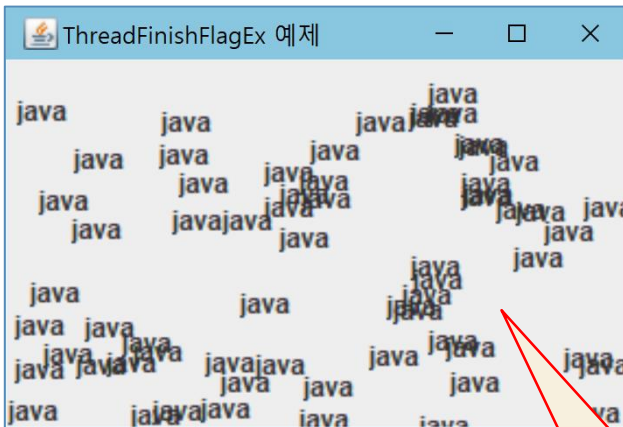
        th.start(); // 스레드 동작시킴
    }
    public static void main(String[] args) {
        new ThreadInterruptEx();
    }
}
```

예제 13-6 : flag를 이용한 스레드 강제 종료

5

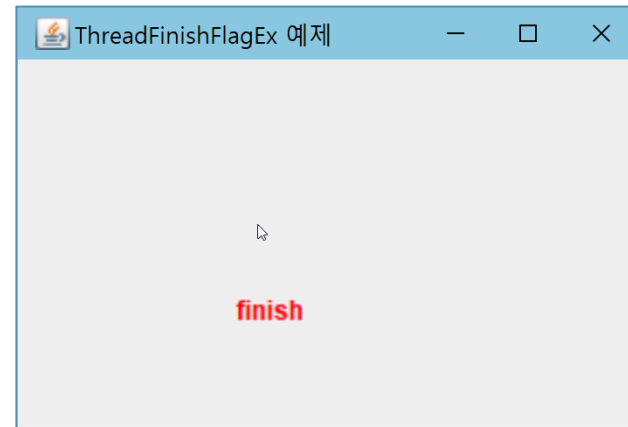
아래 그림과 같이 프로그램이 시작하자마자 0.3초 주기로 "Java" 문자열을 임의의 위치에 생성하는 스레드를 만들어라. 그리고 콘텐츠팬에 마우스를 클릭하면 이 스레드를 종료시키고 "finish" 문자열을 (100, 100) 위치에 출력하라.

➔ 0.5초 주기로 본인의 이름을 임의의 위치에 생성하고, 마우스를 더블 클릭 하는 경우 스레드를 종료 시키고 화면을 clear 한 후 파란색으로 finish를 출력하도록 변경 하시오



스레드가 작동하고, 0.3초 주기로 "Java" 문자열이 임의의 위치에 출력되고 있음

컨텐츠팬에 마우스를 클릭하면 스레드 종료



스레드가 종료하였음

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

```
class RandomThread extends Thread {
    private Container contentPane;
    private boolean flag=false; // 스레드의 종료 명령을 표시하는 플래그. true : 종료 지시
    public RandomThread(Container contentPane) {
        this.contentPane = contentPane;
    }

    void finish() { // 스레드 종료 명령을 flag에 표시
        flag = true;
    }

    @Override
    public void run() {
        while(true) {
            int x = ((int)(Math.random()*contentPane.getWidth()));
            int y = ((int)(Math.random()*contentPane.getHeight()));
            JLabel label = new JLabel("Java"); //새 레이블 생성
            label.setSize(80, 30);
            label.setLocation(x, y);
            contentPane.add(label);
            contentPane.repaint();
            try {
                Thread.sleep(300); // 0.3초 동안 잠을 잔다.
                if(flag==true) {
                    contentPane.removeAll();
                    label = new JLabel("finish");
                    label.setSize(80, 30);
                    label.setLocation(100, 100);
                    label.setForeground(Color.RED);
                    contentPane.add(label);
                    contentPane.repaint();
                    return; // 스레드 종료
                }
            }
            catch (InterruptedException e) {return; }
        }
    }
}
```

```
public class ThreadFinishFlagEx extends JFrame {
    private RandomThread th; // 스레드 레퍼런스

    public ThreadFinishFlagEx() {
        setTitle("ThreadFinishFlagEx 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(null);

        c.addMouseListener(new MouseAdapter() {
            @Override
            public void mousePressed(MouseEvent e) {
                th.finish(); // RandomThread 스레드 종료 명령
            }
        });
        setSize(300,200);
        setVisible(true);

        th = new RandomThread(c); // 스레드 생성
        th.start(); // 스레드 동작시킴
    }

    public static void main(String[] args) {
        new ThreadFinishFlagEx();
    }
}
```

예제 13-6 : wait(), notify()를 이용한 바 채우기

7

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

```
class MyLabel extends JLabel {
    int barSize = 0; // 바의 크기
    int maxBarSize;

    MyLabel(int maxBarSize) {
        this.maxBarSize = maxBarSize;
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.MAGENTA);
        int width = (int)((double)(this.getWidth())
            /maxBarSize*barSize);
        if(width==0) return;
        g.fillRect(0, 0, width, this.getHeight());
    }

    synchronized void fill() {
        if(barSize == maxBarSize) {
            try {
                wait();
            } catch (InterruptedException e) { return; }
        }
        barSize++;
        repaint(); // 바 다시 그리기
        notify();
    }
}
```

```
synchronized void consume() {
    if(barSize == 0) {
        try {
            wait();
        } catch (InterruptedException e) {
            return;
        }
        barSize--;
        repaint(); // 바 다시 그리기
        notify();
    }
}
```

```
class ConsumerThread extends Thread {
    MyLabel bar;

    ConsumerThread(MyLabel bar) {
        this.bar = bar;
    }

    public void run() {
        while(true) {
            try {
                sleep(200);
                bar.consume();
            } catch (InterruptedException e) {
                return;
            }
        }
    }
}
```

```
public class TabAndThreadEx extends
JFrame {
    MyLabel bar = new MyLabel(100);
    TabAndThreadEx(String title) {
        super(title);
        this.setDefaultCloseOperation
            (JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(null);
        bar.setBackground(Color.ORANGE);
        bar.setOpaque(true);
        bar.setLocation(20, 50);
        bar.setSize(300, 20);
        c.add(bar);

        c.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e)
            {
                bar.fill();
            }
        });
        setSize(350,200);
        setVisible(true);

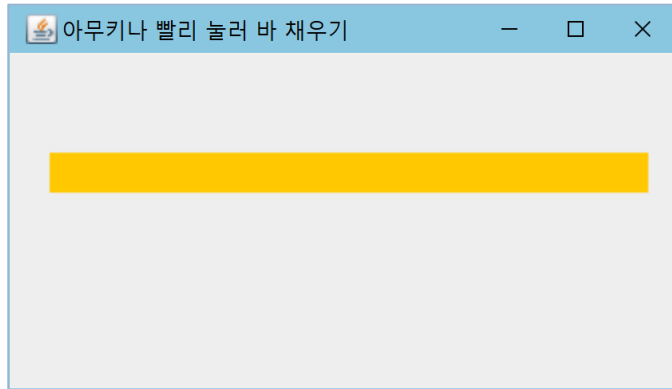
        c.requestFocus();
        ConsumerThread th = new
            ConsumerThread(bar);
        th.start(); // 스레드 시작
    }

    public static void main(String[] args) {
        new TabAndThreadEx(
            "아무키나 빨리 눌러 바 채우기");
    }
}
```

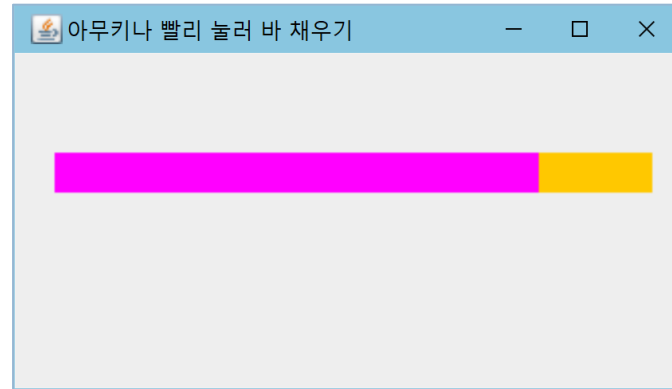
➔ Main에서 300 이하로 maxBarSize를 입력 받아 MyLabel을 생성하도록 수정하시오

실행 결과

8



초기 화면



키를 반복하여 빨리 누른 화면