

Lab Exercise #5

GUI: 이벤트 처리 실습

2018년도 2학기

컴퓨터프로그래밍2

김 영 국

충남대학교 컴퓨터공학과



목차

■ 실습

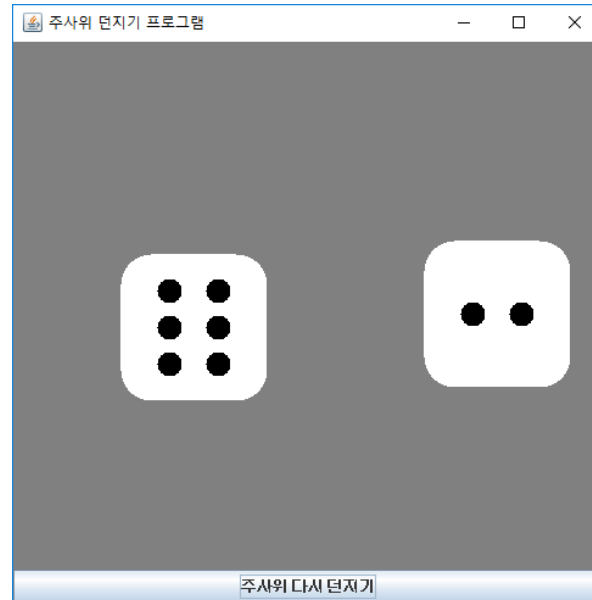
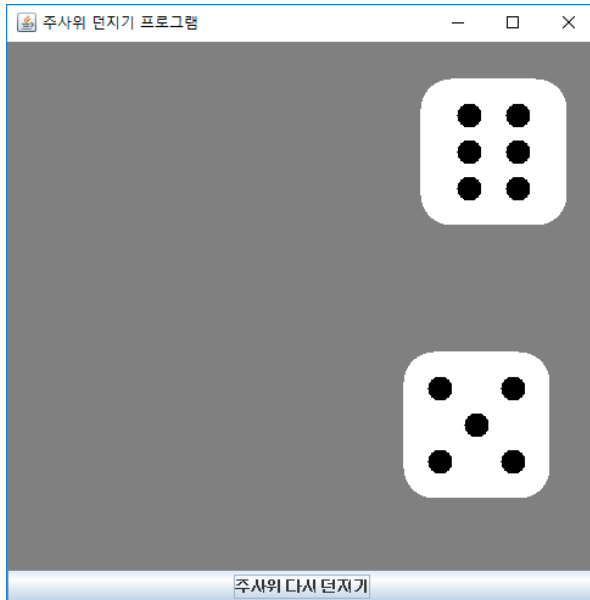
- 주사위 던지기 프로그램
- 마일-킬로 변환 프로그램
- 마우스로 사각형 그리기
- 움직이는 뱀 만들기

■ 과제

- 계산기

실습5-1. 주사위 던지기 프로그램

- 과제 4-2에서 만들었던 프로그램에 이벤트 처리 부분을 추가하여 GUI 프로그램을 완성시켜 보자.
- 프레임 하단에 “주사위 다시 던지기” 버튼을 추가하고, 버튼을 눌렀을 때 화면에 새로운 주사위가 다시 그려지게 만든다.

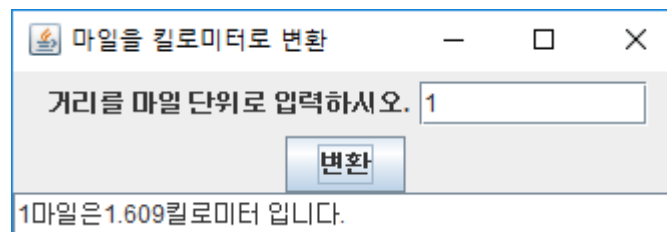


실습5-2. 마일-킬로 변환 프로그램

- 실습3-3에서 만들었던 마일-킬로 변환 프로그램에 이벤트 처리 부분을 추가하여 GUI 프로그램을 완성시켜 보자.
- 아래의 코드를 이전에 작성했던 코드에 추가한다.

```
b1.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        double killo;  
        killo = Double.parseDouble(inputTextField.getText()) * 1.609 ;  
        outputTextField.setText(inputTextField.getText() + "마일은" + String.format("%.3f", killo) + "킬로미터 입니다.");  
    }  
});
```

- 실행 예시는 아래와 같다.





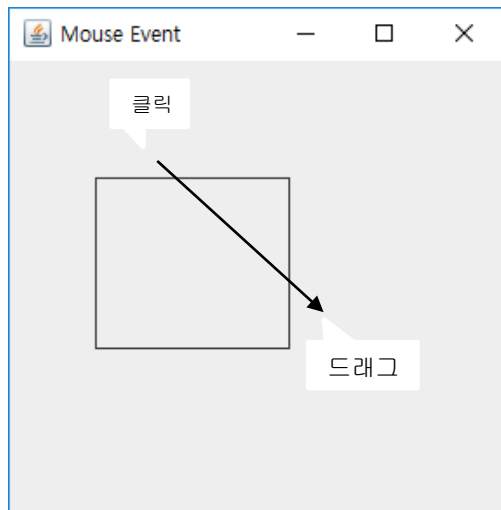
실습5-3. 마우스로 사각형 그리기(1)

- 마우스로 드래그를 한 영역에 사각형이 그려지는 프로그램을 작성해보자.
- 그리고자 하는 패널에 `MouseListener`와 `MouseMotionListener`를 등록한다.
- 마우스의 왼쪽 버튼이 눌렸을 때의 패널의 좌표(x_1, y_1), 드래그가 끝났을 때의 좌표(x_2, y_2)를 사용해 `drawRect()` 메소드의 인자들을 만든다.

실습5-3. 마우스로 사각형 그리기(2)

■ 코드와 실행 예시

```
class MouseEventFrame extends JFrame {  
    MouseEventPanel p = new MouseEventPanel();  
  
    public MouseEventFrame() {  
        setTitle("Mouse Event");  
        setSize(300, 300);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        add(p);  
        setVisible(true);  
    }  
}
```



```
class MouseEventPanel extends JPanel {  
    private int x1, y1, x2, y2;  
    private int smallX, smallY;  
  
    public MouseEventPanel() {  
        // MouseInputAdapter를 사용하면 더 쉽게 구현 가능하다.  
        // 하지만 실습이므로 MouseListener와 MouseMotionListener를 두개다 구현해보자.  
        addMouseListener(new MouseListener() {  
            public void mouseReleased(MouseEvent e) {}  
            public void mousePressed(MouseEvent e) {  
                x1 = e.getX();  
                y1 = e.getY();  
            }  
            public void mouseExited(MouseEvent e) {}  
            public void mouseEntered(MouseEvent e) {}  
            public void mouseClicked(MouseEvent e) {}  
        });  
  
        addMouseMotionListener(new MouseMotionListener() {  
            public void mouseMoved(MouseEvent e) {}  
  
            public void mouseDragged(MouseEvent e) {  
                x2 = e.getX();  
                y2 = e.getY();  
                repaint();  
            }  
        });  
    }  
    @Override  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        smallX = (x1 <= x2) ? x1 : x2;  
        smallY = (y1 <= y2) ? y1 : y2;  
        g.drawRect(smallX, smallY, Math.abs(x1 - x2), Math.abs(y1 - y2));  
    }  
}
```



실습 5-4. 움직이는 뱀 만들기(1)

- 키보드의 방향키를 입력 받아 한 칸씩 움직이는 뱀을 만들어 보자.
- 패널에 `KeyListener()`를 등록한다.
- x좌표의 배열, y좌표의 배열들을 방향키 입력에 따라 값이 바뀌게 만든다.
 - 배열의 i 번째 원소는 $i+1$ 의 원소의 값으로 바뀌고 마지막 원소의 값에는 x , y 에 따라 각각 이동하고 싶은 픽셀만큼의 값을 더해주거나 뺀다.
- `repaint()` 메소드를 호출한다.

주의: 생성자에서 `requestFocus()`, `setFocusable(true)` 메소드를 호출하지 않으면 키보드 입력을 받지 못하기 때문에 꼭 호출해주어야 한다.



실습 5-4. 움직이는 뱀 만들기(2)

- 패널의 필드와 paintComponent() 메소드

```
class KeyEventPanel extends JPanel{

    private int[] xArr = new int[5], yArr = new int[5];
    private int width = 25,height = 25;

    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        for(int i = 0; i < 5; i++) {
            g.drawRect(xArr[i], yArr[i], width, height);
        }
    }
}
```

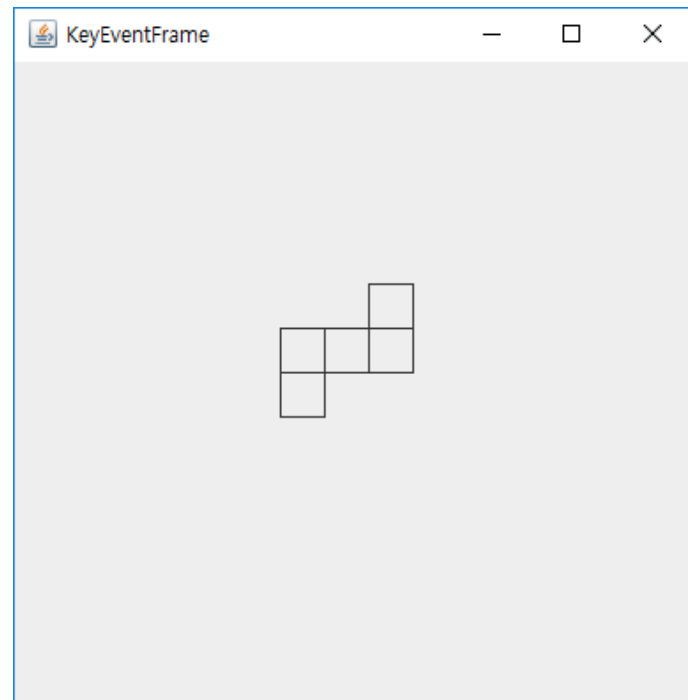
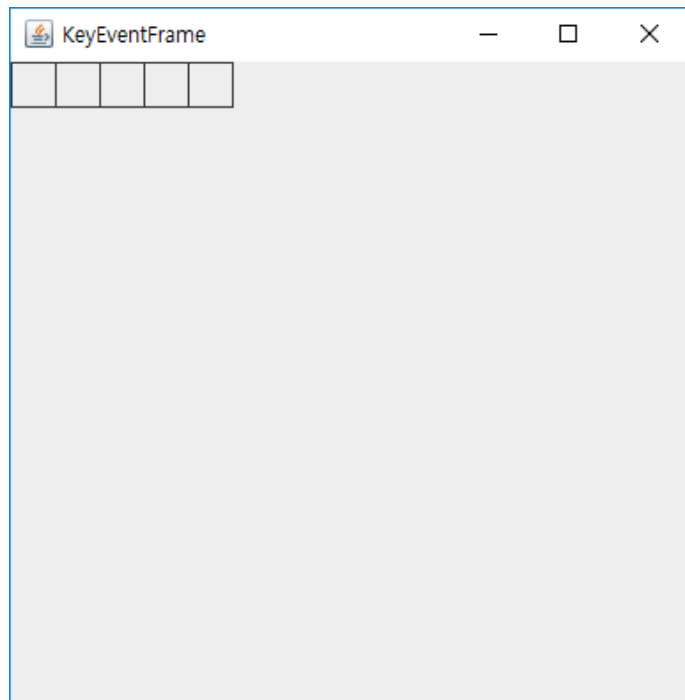

실습 5-4. 움직이는 뱀 만들기(3)

- 패널의 생성자

```
public KeyEventPanel() {
    for(int i = 1; i < 5; i++) {
        xArr[i] = xArr[i - 1] + width;
    }
    addKeyListener(new KeyListener() {
        public void keyTyped(KeyEvent e) {}
        public void keyReleased(KeyEvent e) {}
        public void keyPressed(KeyEvent e) {
            switch (e.getKeyCode()) {
                case KeyEvent.VK_UP:
                    for(int i = 0; i < xArr.length - 1; i++) {
                        xArr[i] = xArr[i + 1];
                        yArr[i] = yArr[i + 1];
                    }
                    yArr[xArr.length - 1] -= height;
                    break;
                case KeyEvent.VK_DOWN:
                    for(int i = 0; i < xArr.length - 1; i++) {
                        xArr[i] = xArr[i + 1];
                        yArr[i] = yArr[i + 1];
                    }
                    yArr[xArr.length - 1] += height;
                    break;
                case KeyEvent.VK_LEFT:
                    for(int i = 0; i < xArr.length - 1; i++) {
                        xArr[i] = xArr[i + 1];
                        yArr[i] = yArr[i + 1];
                    }
                    xArr[xArr.length - 1] -= width;
                    break;
                case KeyEvent.VK_RIGHT:
                    for(int i = 0; i < xArr.length - 1; i++) {
                        xArr[i] = xArr[i + 1];
                        yArr[i] = yArr[i + 1];
                    }
                    xArr[xArr.length - 1] += width;
                    break;
            }
            repaint();
        }
    });
    this.requestFocus();
    setFocusable(true);
}
```

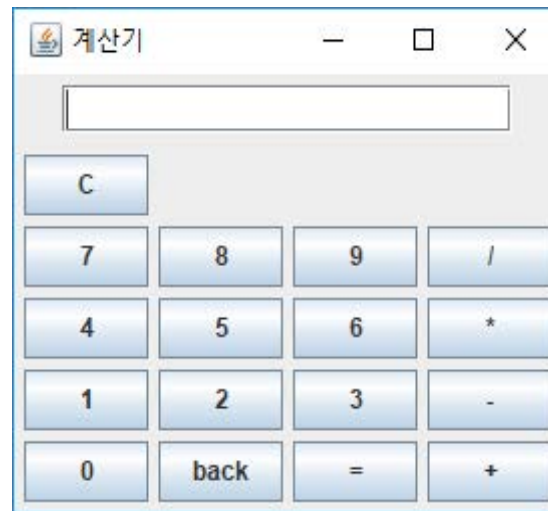
실습 5-4. 움직이는 뱀 만들기(4)

- 실행 예시



과제 5-1. 계산기

- 과제 3-2에서 만들었던 계산기 프로그램에 이벤트 처리 부분을 추가하여 GUI 프로그램을 완성시켜보자.
 - 계산기 버튼들에 대하여 모든 기능이 작동하도록 만들어야 한다.
 - 0으로 나누는 경우에 대하여 try-catch 구문을 사용해 예외처리가 되어 있어야 한다.
- 실행 예시





과제 제출 및 기한

- 제출 방법
 - 사이버캠퍼스를 통하여 제출
 - 소스코드를 제출
- 제출 기한
 - 이번주 토요일(10/13) 자정