

중간고사 온라인 과제

과목명	JAVA 프로그래밍 실습	담당 교수	안용학
학과	인공지능융합	학번	2019313464
성명	안예림		

목차

1. 문제1 : 가위바위보

1.1 사전 작업

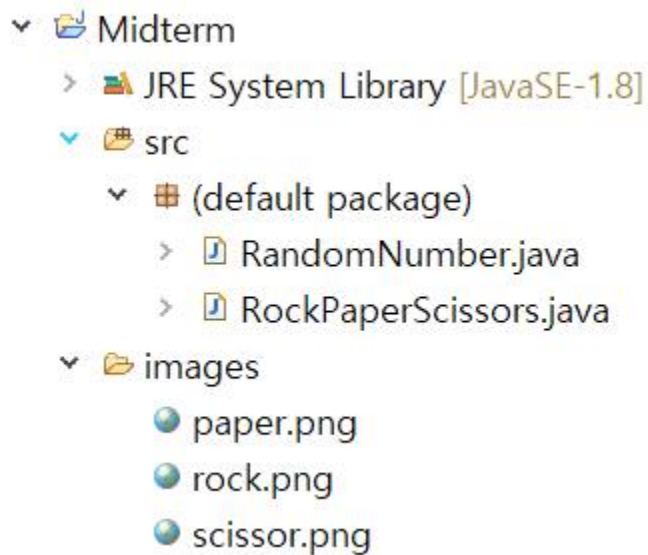
1.2 코드 설명

1.3 실행 결과

2. 문제2 : 랜덤 숫자 2개

2.1 코드 설명

2.2 실행 결과



Midterm이라는 파일을 만들어 두 문제를 모두 이 안에 넣었다. RandomNumber.java는 문제 2의 소스코드 파일이고, RockPaperScissors.java는 문제 1의 소스코드 파일이다.

paper, rock, scissor.png 파일에는 이미지가 들어있다.

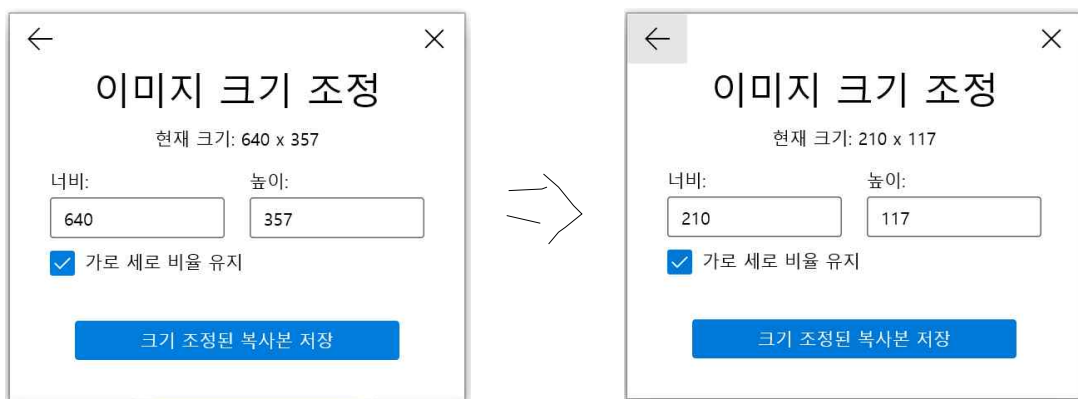
1. 다음과 같은 가위바위보 게임 JAVA Application을 작성하시오. (50점)

제출 파일 : RockPaperScissor.java

1.1 사전 작업

가위바위보 게임 Java application을 만들기 위해서 먼저 가위바위보에 해당하는 이미지 파일이 필요했다.

구글에서 동일한 가위바위보 사진 파일을 다운 받았다. rock.png는 rock(바위), paper.png는 paper(보), scissor.png는 scissor(가위)에 해당하도록 저장했다. 그러나 사진의 크기가 너무 컸다. 640 x 357이어서 크기 조정이 필요했다. 그래서 210 x 117 정도로 크기를 조정하고



다시 사진을 저장했다.

이 3개의 이미지를 이클립스의 Midterm 자바 프로젝트에 images 파일을 만들어 그 안에 copy해주었다.

1.2 코드 설명

```
RandomNumber.java  RockPaperScissors.java  *NestedPanel.java
1=import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import java.util.Random;
5
6 import javax.swing.ImageIcon;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JPanel;
10 import javax.swing.SwingConstants;
11
12 public class RockPaperScissors {
13
14=    public static void main(String[] args) {
15
16        JFrame frame = new JFrame("Rock-Paper-Scissors");
17        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18
19        JPanel primaryPanel = new JPanel();
20        primaryPanel.setBackground(Color.white);
21        primaryPanel.setPreferredSize(new Dimension(315, 190));
22        primaryPanel.setLayout(null);
```

1~10 : 필요한 모듈을 import해주었다.

12 : Class의 이름은 RockPaperScissor로 영어로 가위바위보를 적어주었다.

14 : Class를 만들 때 main 함수를 추가해주었다.

16~17 : JFrame을 통해 프레임을 만들어 주고, 프레임 이름은 "Rock-Paper-Scissors"로 해

주었다. frame을 종료했을 때 프로그램이 종료될 수 있도록 넣어주었다.

19~22 : 프레임에 add할 패널을 만들어주었다. 패널의 색깔은 하얀색이고, 가로 315, 세로 190의 크기로 설정해주었고, layout은 null로 설정했다.

```
23
24     Font fnt = new Font("Century", Font.BOLD, 15);
25
26     JLabel User1, User2, Vs, Pic1, Pic2, Result;
27
28     User1 = new JLabel("USER1");
29     User1.setFont(fnt);
30     User1.setBounds(50, -40, 100, 100);
31     primaryPanel.add(User1);
32
33     User2 = new JLabel("USER2");
34     User2.setFont(fnt);
35     User2.setBounds(210, -40, 100, 100);
36     primaryPanel.add(User2);
37
38     Vs = new JLabel("VS");
39     Vs.setFont(fnt);
40     Vs.setBounds(145, 90, 50, 50);
41     primaryPanel.add(Vs);
42
```

24 : Century라는 폰트에 볼드체를 넣어주고, 15포인트로 설정했다.

26 : 라벨들을 설정해준다.

28~31 : 먼저 왼쪽 위에 USER1이라고 새로운 라벨을 설정하고, 폰트는 아까 만든 폰트로 설정해준다. 위치는 실행해가며 적절한 위치를 설정해주었다. 그리고 아까 만든 패널에 add시켜 준다.

33~36 : 마찬가지로 방법으로 오른쪽 위에 나타낼 USER2 라벨을 만들어주었다.

38~41 : 마찬가지로 방법으로 중간에 들어간 VS 라벨을 만들고, 적절한 위치에 넣어준다.

```
43     ImageIcon[] images = { // ImageIcon 클래스 객체 images를 배열로 선언
44         new ImageIcon("images/rock.png"), // [0]
45         new ImageIcon("images/paper.png"), // [1]
46         new ImageIcon("images/scissor.png"), // [2]
47     };
48
49     Random generator = new Random();
50     int ran1 = generator.nextInt(3);
51     int ran2 = generator.nextInt(3);
52
53     Pic1 = new JLabel(images[ran1], SwingConstants.CENTER);
54     Pic1.setBounds(10, 30, 120, 120);
55     primaryPanel.add(Pic1);
56
57     Pic2 = new JLabel(images[ran2], SwingConstants.CENTER);
58     Pic2.setBounds(180, 30, 120, 120);
59     primaryPanel.add(Pic2);
60
```

43~47 : ImageIcon 클래스의 객체 배열로 images 파일에 넣은 3개의 이미지를 저장해주었다. 바위는 0번째에, 보는 1번째에, 가위는 2번째에 들어간 것을 알 수 있다.

49~51 : 가위바위보 이미지가 실행할 때 마다 랜덤하게 설정되도록 random generator를 만들고, 정수 변수인 ran1, ran2에 0,1,2 중 하나의 값을 랜덤으로 받아 대입해주었다.

53~55 : USER1의 가위바위보 이미지를 Pic1라벨에 넣어주고, 적절한 위치에 bound 시켜주고, 패널에 add해준다.

57~59 : USER2의 가위바위보 이미지를 Pic2라벨에 넣어주고, 적절한 위치에 bound 시켜주고, 패널에 add해준다.

```

61 // 바위 : 0, 보 : 1, 가위 : 2
62 String win = new String();
63 if (ran1 == ran2)
64     win = "End in a tie!!";
65 else if (ran1 == 0 && ran2 == 2)
66     win = "USER1 WIN!!";
67 else if (ran1 == 2 && ran2 == 0)
68     win = "USER2 WIN!!";
69 else if (ran1 > ran2)
70     win = "USER1 WIN!!";
71 else
72     win = "USER2 WIN!!";
73
74 Result = new JLabel(win);
75 Result.setFont(fnt);
76 Result.setBounds(110, 120, 300, 100);
77 primaryPanel.add(Result);
78

```

61 : 바위(0), 보(1), 가위(2) 라는 것을 생각해보면 1은 0을 이기고, 2는 1을 이긴다. 그러나, 0이 2를 이긴다. 그래서 0이 2를 이기는 것만 먼저 설정을 해주고, 나머지(else)는 숫자가 큰 쪽이 이긴다고 설정해주기로 했다.

62 : 누가 이겼는지를 보여주는 문자열을 생성했다.

63~64 : 비겼을 경우에 "End in a tie!!"를 win변수에 대입한다.

65~68 : 바위(0)와 가위(2)가 만났을 경우에 바위 쪽이 이기는 문자열을 win에 대입한다.

69~72 : 나머지 경우에는 숫자가 큰 쪽이 이기는 것으로 설정한다.

74~77 : 결과를 보여주는 label인 result를 만들어 win 문자열을 넣어준다. 폰트를 설정하고, 적절한 위치에 바운드시켜주고, 패널에 add해준다.

```

79 frame.getContentPane().add(primaryPanel);
80 frame.pack();
81 frame.setVisible(true);
82 } // main
83
84 } // RockPaperScissors
85

```

79~81 : 마지막으로 만들었던 프레임을 pack해주고, 볼 수 있도록 만들어준다.

82 : main함수 끝

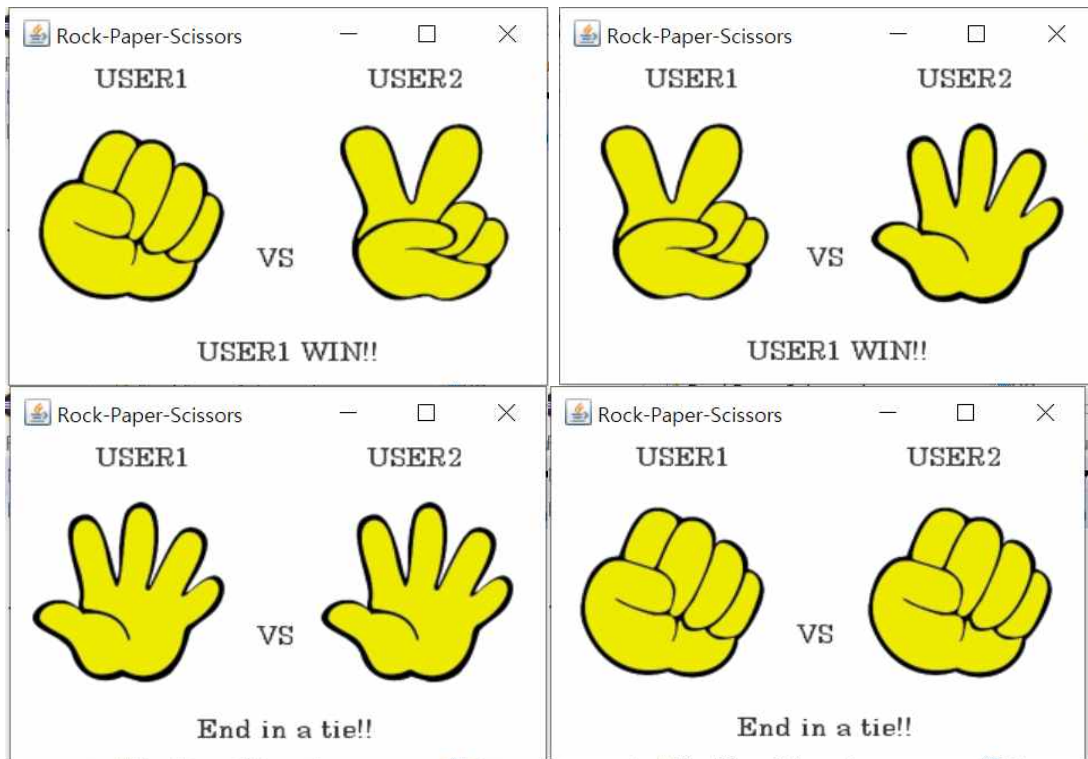
84 : RockPaperScissors 클래스 끝

1.3 실행 결과

이미지가 랜덤하게 나오기 때문에 여러 번 실행해주어 결과가 잘 나왔는지 확인해주었다.



여러 경우에서 잘 실행되었음을 알 수 있다.



2. 다음과 같은 JAVA Application을 작성하시오. (50점)

제출 파일 : RandomNumber.java

2.1 코드 설명

```
RandomNumber.java
1=import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import java.util.Random;
5
6 import javax.swing.JFrame;
7 import javax.swing.JLabel;
8 import javax.swing.JPanel;
9
10 public class RandomNumber {
11
12     public static void main(String[] args) {
13
14         JFrame frame = new JFrame("MIDTERM EXAMINATION");
15         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16
17         JPanel primaryPanel = new JPanel();
18         primaryPanel.setBackground(Color.white);
19         primaryPanel.setPreferredSize(new Dimension(440, 440));
20         primaryPanel.setLayout(null);
21
```

1~8 : 필요한 모듈 import

10 : Class 이름은 RandomNumber로 해주었다.

12 : Class를 만들 때 main을 추가해주었다.

14~15 : 프레임을 만들어주고, 프레임의 이름은 MIDTERM EXAMINATION으로 해주었다.
(실행 예랑 동일), 프레임을 종료하면 프로그램이 종료되도록 설정했다.

17~20 : 실행 예를 보면 하얀색 패널이 배경으로 있고, 그 안에 색상이 들어간 패널이 10 정도의 둘레를 끼고 있는 것을 볼 수 있다. 따라서 기본 패널을 만들어주었다. 위아래 왼쪽 오른쪽 10 여유를 두고, 440,440 크기로 설정해주고, 하얀색으로 설정해주었다.

```
22         Random generator = new Random();
23
24         int r, g, b;
25         r = generator.nextInt(256);
26         g = generator.nextInt(256);
27         b = generator.nextInt(256);
28
29         JPanel colorPanel = new JPanel();
30         colorPanel.setBackground(new Color(r, g, b));
31         colorPanel.setBounds(10, 10, 420, 420);
32         colorPanel.setLayout(null);
33         primaryPanel.add(colorPanel);
34
35         int num1 = generator.nextInt(6) + 1;
36         int num2 = generator.nextInt(6) + 1;
37
```

22 : 패널 색상을 랜덤으로 정해주어야 하므로 Random을 설정해주었다.

24~27 : 패널 색상을 r,g,b 값으로 정해주기 위해 0~255에 해당하는 정수를 각각 랜덤으로

뽑아 대입해준다.

29~33 : color가 랜덤으로 들어갈 colorPanel을 새로 만들어준다. JPanel 객체의 배경색 (Background) 색상은 랜덤으로 생성된 r,g,b값을 넣어 색깔을 만들어준다. 크기는 420x420으로 해주고, layout은 null로 한다. 프로그램이 실행될 때마다 다른 색을 가지도록 된다. 마지막으로 아까 만든 primaryPanel에 add해준다.

35~36 : 이제 colorPanel에 들어갈 숫자 2개를 랜덤으로 만들어준다. num1,num2에 각각 대입된다.

```
38      JLabel n1, n2;
39      n1 = new JLabel(Integer.toString(num1));
40      n2 = new JLabel(Integer.toString(num2));
41
42      if (num1 > num2) {
43          n1.setFont(new Font("Verdana", Font.BOLD, 100));
44          n2.setFont(new Font("Verdana", Font.BOLD, 40));
45      }
46      else if (num1 < num2) {
47          n1.setFont(new Font("Verdana", Font.BOLD, 40));
48          n2.setFont(new Font("Verdana", Font.BOLD, 100));
49      }
50      else {
51          n1.setFont(new Font("Verdana", Font.BOLD, 70));
52          n2.setFont(new Font("Verdana", Font.BOLD, 70));
53      }
```

38~40 : JLabel 객체 n1, n2를 만들고, 랜덤으로 만든 숫자값을 텍스트로 가지도록 해준다. 이 과정에서 정수를 문자열로 바꿔주었다. 그냥 정수값으로 넣으면 에러가 난다.

42~45 : 만약 n1에 들어있는 숫자값이 n2에 들어있는 숫자값보다 클 경우, n1의 크기를 100(크게), n2의 크기를 40(작게)으로 설정해주었다. Verdana 폰트에 볼드체를 사용했다.

46~48 : 만약 n2에 들어있는 숫자값이 n1에 들어있는 숫자값보다 클 경우, n2의 크기를 100(크게), n1의 크기를 40(작게)으로 설정해주었다.

49~52 : 그 외의 경우(두 숫자값이 같을 경우), 동일한 크기인 70으로 설정해주었다.

```
54      int r1, g1, b1, w1, h1;
55      r1 = generator.nextInt(256);
56      g1 = generator.nextInt(256);
57      b1 = generator.nextInt(256);
58      w1 = generator.nextInt(420 - 100);
59      h1 = generator.nextInt(420 - 100);
60
61      int r2, g2, b2, w2, h2;
62      r2 = generator.nextInt(256);
63      g2 = generator.nextInt(256);
64      b2 = generator.nextInt(256);
65      w2 = generator.nextInt(420 - 100);
66      h2 = generator.nextInt(420 - 100);
67
68      n1.setForeground(new Color(r1, g1, b1));
69      n1.setBounds(w1, h1, 100, 100);
70      colorPanel.add(n1);
71
72      n2.setForeground(new Color(r2, g2, b2));
73      n2.setBounds(w2, h2, 100, 100);
74      colorPanel.add(n2);
75
```

54~59 : JLabel의 객체인 n1(첫번째 숫자)의 전경색(Foreground color)와 위치(position)을 랜덤하게 설정해주기 위해 전경색 r1,g1,b1 변수를 0~255사이 값으로 나타낼 수 있도록 했다. 또한 420x420크기인 패널 안에서 랜덤한 위치를 찾기 위해 0~419 사이 정수값을

weight(w1), height(h1)에 넣으려고 했다. 그러나 극단적인 경우(420,420에 가깝도록)가 나오는 경우, 패널 안에 숫자가 나타나지 않는 문제가 발생했다. 패널 밖에 나타나 보이지 않는 것이다. 따라서, 가장 큰 폰트인 100(다른 숫자보다 큰 경우)가 화면 안에 나타날 수 있도록 0~320(420-100) 사이의 정수 값을 w1,h1에 넣어주었다. 그럼 가장 극단적인 경우인 320,320 포인트에 n1의 숫자값이 n2보다 커서 폰트가 100일 때에도 화면 안에 숫자가 보인다.

61~66 : JLabel의 객체인 n2(두번째 숫자)의 전경색과 위치를 랜덤하게 설정해주기 위해 r2,g2,b2와 w2,h2를 사용했다. n1과 마찬가지로 방법으로 만들었다.

68~70 : n1에 랜덤으로 만들어진 전경색을 넣어주고, 랜덤으로 만들어진 위치에 bound 시켜 준다. 가로 100, 세로 100으로 폰트 크기가 100일 때도 잘리지 않고 나타날 수 있도록 했다. n1을 colorPanel에 add해준다.

72~74 : 동일한 방법으로 n2의 전경색과 위치도 설정해주었다. n2을 colorPanel에 add해준다.

```

75
76     frame.getContentPane().add(primaryPanel);
77     frame.pack();
78     frame.setVisible(true);
79 } // main
80
81 } // RandomNumber

```

76~78 : 프레임에 패널이 add될 수 있게 해주고, frame이 나타날 수 있도록 pack해주고, 보이도록 해주었다.

79 : main 끝

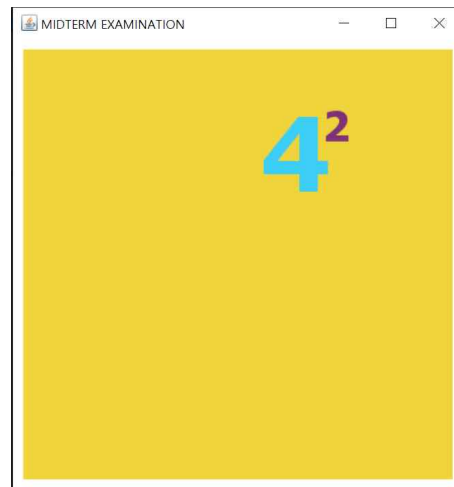
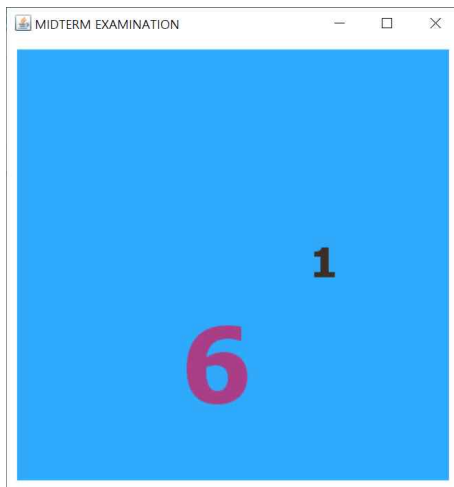
81 : RandomNumber class 끝

2.2 실행 결과

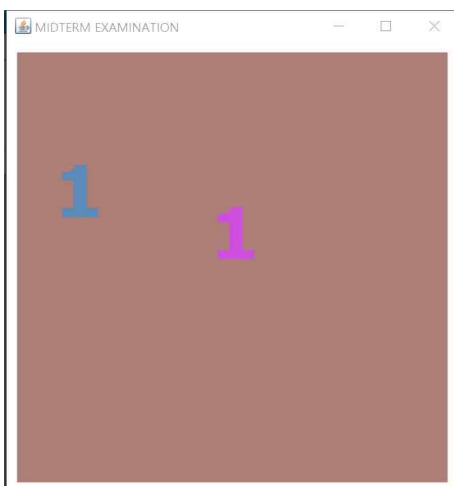
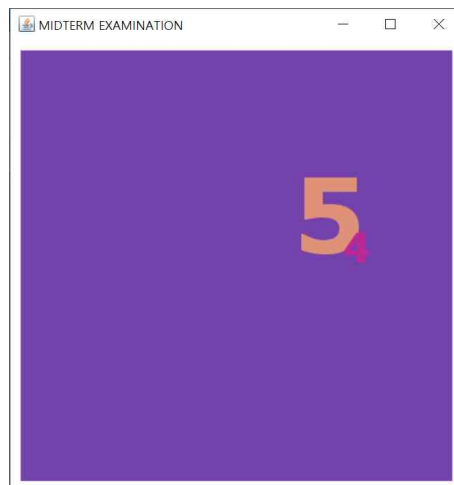
JPanel 객체의 배경색, JLabel의 2개의 객체의 전경색, 위치 모두 다 랜덤으로 설정되므로, 여러 번 실행해보았으며, 여러 실행화면을 캡처했다.

-폰트 크기 100와 40이 정상적으로 출력된 모습





- 동일한 크기인 70이 보여지는 모습 / 숫자가 약간 겹쳐져서 나타나는 모습



감사합니다 !