

# 목차

---

1. 클래스 연습문제
2. JFrame

# 연습문제1

---

## ▣ Data 클래스를 생성하고, 사용해 보세요.

- day, month, year를 멤버변수로 가짐
- 각 멤버변수에 대해 get, set함수를 가짐
  - ▶ get 함수는 정보를 가져오는 함수
  - ▶ set 함수는 정보를 저장하는 함수
- 각 멤버변수를 입력으로 받는 생성자를 가짐
- 유효한 날짜인지를 판단하는 isValid 함수를 가짐

## ▣ isValid함수를 사용하여 각 케이스에 대해 유효한 날짜인지 판단하세요

- 2000년 2월 30일
- 2020년 2월 10일

## 연습문제2

---

### ▣ 다음 멤버를 가지고 직사각형을 표현하는 Rectangle 클래스를 작성하세요.

- int 타입의 x, y, width, height 필드: 사각형을 구성하는 점과 크기 정보
- x, y, width, height 값을 매개변수로 받아 필드를 초기화하는 생성자
- int square(): 사각형 넓이 반환
- void show(): 사각형의 좌표와 넓이를 화면에 출력
- boolean contains(Rectangle r): 매개변수로 받은 r이 현재 사각형 안에 있으면 true 반환

### ▣ 인터페이스 구현

- (2,2,8,7), (5,5,6,6), (1,1,10,10) 세 개의 사각형 생성
- 첫번째 사각형의 좌표와 (HxW) 출력
- 두번째 사각형의 면적 출력
- 세번째 사각형이 첫번째와 두번째 사각형을 포함하는지 출력

## 연습문제3 - 숫자야구

---

### ■ 숫자야구란?? 1~9까지의 숫자 중 임의의 3개의 숫자를 정한 뒤 상대가 숫자를 맞추는 게임

- 매 회마다 상대는 예측한 숫자를 부르고, 출제자는 숫자가 얼마나 일치하는지 결과를 알려줍니다. 결과 도출은 다음과 같이 합니다.
  - ▶ 숫자는 맞지만, 위치가 틀렸을 때 볼
  - ▶ 숫자와 위치가 모두 맞았을 때 스트라이크
  - ▶ 숫자와 위치 모두 틀렸을 때 아웃
- 출제자는 볼, 스트라이크, 아웃의 개수를 알려줍니다.
- 3 스트라이크의 경우, 상대가 이깁니다.
- 10회의 턴에도 상대가 맞추지 못할 경우 출제자가 이깁니다.

# 연습문제3 - 숫자야구

■ NumberBaseball 클래스를 생성하세요. 출제자는 프로그램, 맞추는 상대는 사용자가 됩니다.

- int 타입의 필드: 사용자에게 입력 받을 숫자, 정답 숫자
- 정답 숫자를 랜덤하게 생성하는 생성자
  - ▶ 1~9사이 3개 숫자를 겹치지 않게 랜덤하게 생성
  - ▶ Math.random() 사용
- void play: 사용자에게 1~9사이 3개의 숫자를 겹치지 않게 입력 받음
- boolean check: 맞춘 결과를 예측 하고 출력해주는 함수

## ■ 인터페이스 구현

- 맞추는 과정을 10번 반복하고 10번 이내에 정답을 맞추면 win을, 정답을 못 맞추면 lose를 출력한다.

[1번째 시도입니다.]

1 ~ 9 사이의 3자리수를 입력해주세요.  
1번째 숫자: 1  
2번째 숫자: 22  
범위를 초과하였습니다. 다시 입력해주세요  
2번째 숫자: 2  
3번째 숫자: 2  
중복 되었습니다. 다시입력해주세요

1 ~ 9 사이의 3자리수를 입력해주세요.  
1번째 숫자: 2  
2번째 숫자: 1  
3번째 숫자: 3  
입력한 숫자는 2, 1, 3입니다.  
0스트라이크, 1볼입니다.

[2번째 시도입니다.]

1 ~ 9 사이의 3자리수를 입력해주세요.  
1번째 숫자: 4  
2번째 숫자: 5  
3번째 숫자: 6  
입력한 숫자는 4, 5, 6입니다.  
0스트라이크, 2볼입니다.

# Jframe

---

## ■ Jframe으로부터 상속받아서 사용

```
public class MyFrame extends JFrame {
    MyFrame () {
        setTitle("첫번째 프레임");
        setSize(300, 400);
        // 프레임을 닫았을 때 프로그램이 종료되도록 설정
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // 프레임을 화면 가운데에 배치
        setLocationRelativeTo(null);
        // 레이아웃 설정
        getContentPane().setLayout(new GridLayout(10, 2));
        setVisible(true);
    }

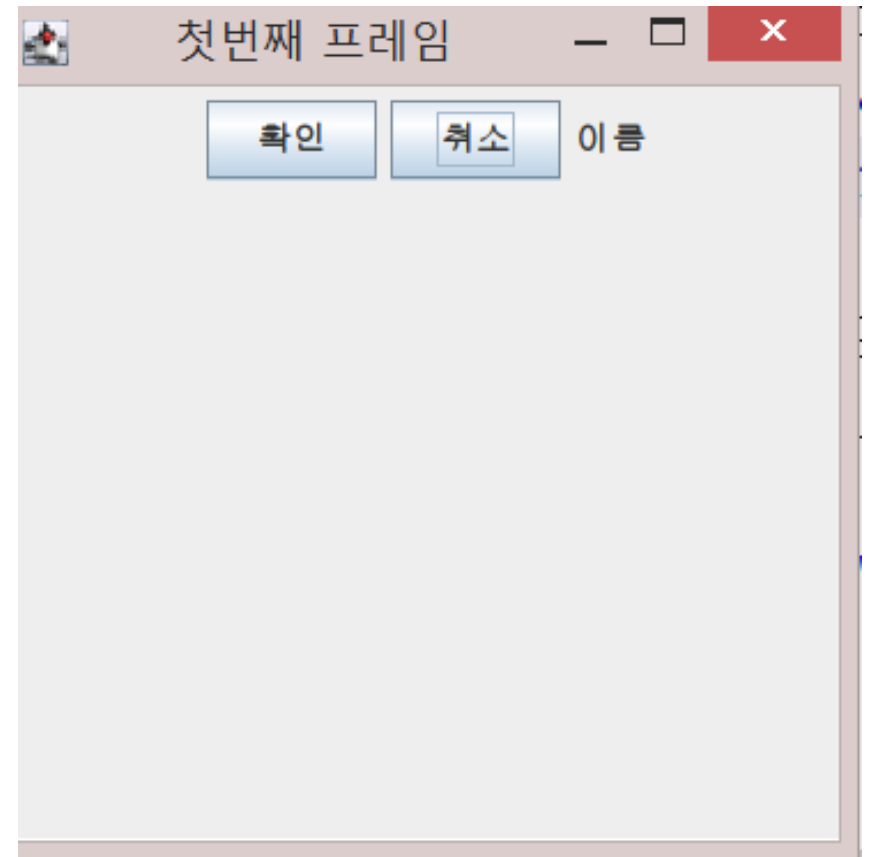
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new MyFrame();
    }
}
```

# Jframe

---

## ■ Container에 컴포넌트들 추가

```
Container c = getContentPane();  
c.setLayout(new FlowLayout());  
  
JButton j = new JButton("확인");  
JButton j2 = new JButton("취소");  
  
c.add(j);  
c.add(j2);  
  
JLabel l1 = new JLabel("이름");  
c.add(l1);
```

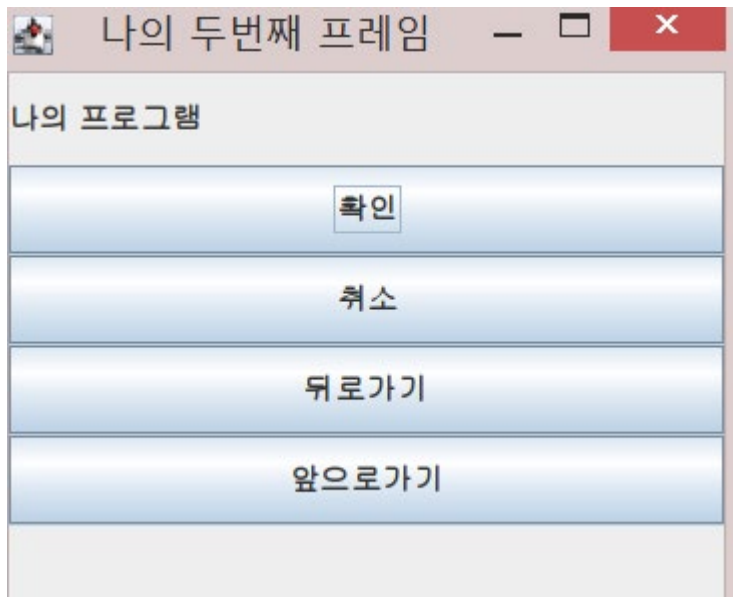


# Jframe

---

## ▣배열을 사용하여 컴포넌트 생성가능

```
public class MyFrameEx extends JFrame{  
    Container c;  
    JButton button_lise[] = new JButton[4];  
    String str_list[] = {"확인", "취소", "뒤로가기", "앞으로가기"};  
  
    c=getContentPane();  
    JLabel la = new JLabel("나의 프로그램");  
    c.add(la);  
    for(int i =0; i< 4; i++) {  
        button_lise[i]= new JButton(str_list[i]);  
        c.add(button_lise[i]);  
    }  
}
```





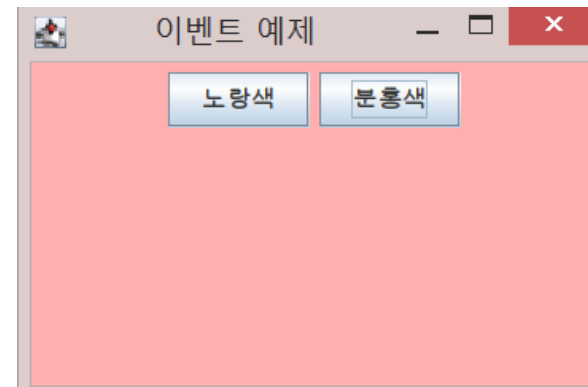
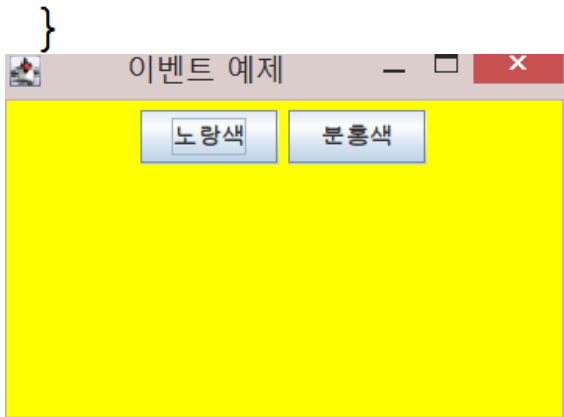
# Jframe

---

## ■ 예시: 버튼을 누르면 색이 바뀌는 코드

- ActionListener를 사용하여 버튼에 대한 이벤트 처리

```
class MyEvent extends JFrame implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == b1) {
            p.setBackground(Color.YELLOW);
        } else if (e.getSource() == b2) {
            p.setBackground(Color.pink);
        }
    }
}
```



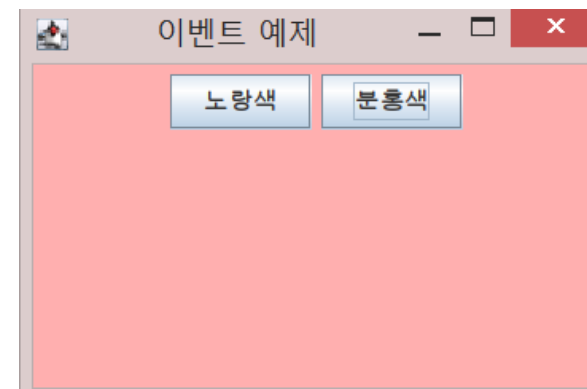
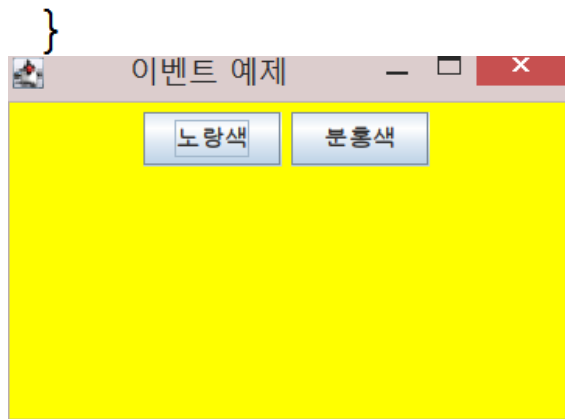
# Jframe

---

## ■ 예시: 버튼을 누르면 색이 바뀌는 코드

- ActionListener를 사용하여 버튼에 대한 이벤트 처리

```
class MyEvent extends JFrame implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == b1) {
            p.setBackground(Color.YELLOW);
        } else if (e.getSource() == b2) {
            p.setBackground(Color.pink);
        }
    }
}
```



## 연습문제3-2

---

▣ 앞에서 구현한 숫자야구를 JFrame으로 구현해보세요.