

# 자바프로그래밍2

## 11주차 실습

스윙 컴포넌트

# 실습 평가 방법

- 점수 : 100점 만점 기준(문제 별 난이도에 따라 점수 부여)
- 채점 기준 : 완성도, 작동 유무, 일부 오류 등에 따라 감점
  - 프로그램이 동작하지 않거나, 코드 공유, ChatGpt 사용 등의 부정행위 적발 시 0점
  - 소스 코드에 허점, 잘못된 들여쓰기, 일부 입출력 오작동 시 정도에 따라 감점
- 제출 기한 : 실습 당일 23시 59분까지(이후 제출 불가능)
  - 실습 시간(14:00-15:50) 내 제출 시 감점X
  - 18:00 까지 제출 시 채점 점수의 5% 감점
  - 20:00 까지 제출 시 채점 점수의 10% 감점
  - 23:59 까지 제출 시 채점 점수의 20% 감점

# 실습 제출 방법

- 압축 파일명 : [n주차\\_학번\\_이름.zip](#)
- 소스 파일 : Eclipse에서 Export한 zip 파일 내 소스 파일(.java)
- 보고서 : 각 문제별 문제 번호 및 소스 코드 실행 결과 화면 캡처한 **pdf 파일** - **부재 시 감점**
- 소스 파일과 보고서를 압축하여 주차별로 위 압축 파일명과 같이 e-루리에 제출
- e-루리 접속 오류 등 특별한 사유로 인해 제출하지 못하는 경우
  - [rkdwlgh01@naver.com](mailto:rkdwlgh01@naver.com) 해당 e-mail을 통해 제출

# 실습 조교 및 질의응답

- e-루리 Q&A 게시판 활용 (작성 후 e-루리 메시지 시 빠른 응답 가능)
- 실습 TA의 e-mail 활용
  - 강지호 : [rkdwlgh01@naver.com](mailto:rkdwlgh01@naver.com)

# [자바프로그래밍2] 11주차 실습 문제

제출 기한 11.14(목) 23:59 전까지

# 문제 1 (40점)

- 가위바위보 게임을 **RPSGame**에서 다음 조건에 맞게 구현하고 테스트하세요.

- 플레이어는 가위, 바위, 보 버튼 중 하나를 선택하고 화면에 표시
- 컴퓨터는 **랜덤**으로 가위, 바위 보 중 하나를 선택하고 화면에 표시
- 두 선택에 따른 결과를 화면에 표시
- 결과에 따라 점수를 부여하고 **받은 점수와 총 점수**를 화면에 표시
  - 승리 : 랜덤으로 1~10점 **획득**
  - 패배 : 랜덤으로 1~10점 **차감**
  - 무승부 : 점수 변화 없음
- 승, 무, 패 기록을 화면에 표시 (ex. 1승 3무 2패)
- 모든 가위, 바위, 보는 텍스트가 아닌 **이미지**로 표시
- restart 버튼을 생성하여 게임 다시 시작(즉, 초기 화면으로 돌아감)

- 클래스 명 : **RPSGame** 📄 RPSGame.java

# 문제 1 (40점)

```
public class RPSGame extends JFrame implements ActionListener {
    private JLabel playerChoiceLabel, computerChoiceLabel, scoreLabel, resultLabel, totalLabel;
    private JButton rockButton, paperButton, scissorsButton, restartButton;
    private ImageIcon rockIcon, paperIcon, scissorsIcon;
    private int wins = 0, losses = 0, draws = 0, score = 0;
    private String[] choices = {"가위", "바위", "보"};
    private Random random = new Random();

    public RPSGame() {
        setTitle("가위바위보 게임");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout(0, 25));

        // Result Panel
        JPanel resultPanel = new JPanel();
        resultPanel.setLayout(new GridLayout(1, 4));

        scoreLabel = new JLabel("Score: 0", SwingConstants.CENTER);
        resultLabel = new JLabel("", SwingConstants.CENTER);
        totalLabel = new JLabel("0승 0무 0패", SwingConstants.CENTER);

        restartButton = new JButton("restart");
        restartButton.addActionListener(e -> restart());

        resultPanel.add(totalLabel);
        resultPanel.add(resultLabel);
        resultPanel.add(scoreLabel);
        resultPanel.add(restartButton);

        add(resultPanel, BorderLayout.NORTH);

        // Choice Panel
        JPanel choicePanel = new JPanel();
        choicePanel.setLayout(new GridLayout(1, 2));

        JPanel playerPanel = new JPanel(new BorderLayout());
        playerPanel.add(new JLabel("플레이어", SwingConstants.CENTER), BorderLayout.NORTH);
        playerChoiceLabel = new JLabel("", SwingConstants.CENTER);
        playerPanel.add(playerChoiceLabel, BorderLayout.CENTER);

        JPanel computerPanel = new JPanel(new BorderLayout());
        computerPanel.add(new JLabel("컴퓨터", SwingConstants.CENTER), BorderLayout.NORTH);
        computerChoiceLabel = new JLabel("", SwingConstants.CENTER);
        computerPanel.add(computerChoiceLabel, BorderLayout.CENTER);

        choicePanel.add(playerPanel);
        choicePanel.add(computerPanel);

        add(choicePanel, BorderLayout.CENTER);
    }
}
```

```
// Button Panel
JPanel buttonPanel = new JPanel();

rockIcon = createIcon("imgs/rock.png");
paperIcon = createIcon("imgs/paper.png");
scissorsIcon = createIcon("imgs/scissors.png");

rockButton = createButton(rockIcon);
paperButton = createButton(paperIcon);
scissorsButton = createButton(scissorsIcon);

buttonPanel.add(rockButton);
buttonPanel.add(paperButton);
buttonPanel.add(scissorsButton);

add(buttonPanel, BorderLayout.SOUTH);

// TODO : 버튼 패널 Border 설정

setVisible(true);
}

private ImageIcon createIcon(String imagePath) {
    ImageIcon icon = new ImageIcon(imagePath);
    Image image = icon.getImage().getScaledInstance(100, 100, java.awt.Image.SCALE_SMOOTH);
    icon = new ImageIcon(image);
    return icon;
}

private JButton createButton(ImageIcon icon) {
    // TODO : 버튼 생성
}

private void restart() {
    // TODO : 초기 화면
}


// TODO : 그외 필요한 코드 작성

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

# 문제 1 (40점)

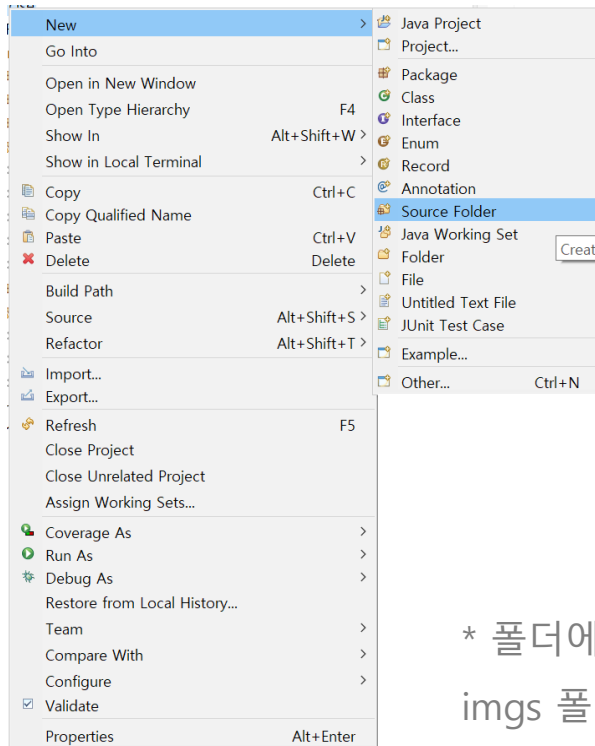
## ➤ 사용할 이미지 프로젝트에 추가

프로젝트 우클릭 → New → Source Folder → 폴더명은 imgs

 imgs

해당 폴더에 이미지 파일 추가(rock.png, scissors.png, paper.png)

파일 경로는 "imgs/rock.png"



\* 폴더에 파일 추가

imgs 폴더 우클릭 → import → General → File System → 파일이 있는 폴더 선택 → 추가할 파일 모두 선택



# 문제 1 (40점)

실행결과 예시 :



## 문제 2 (60점)

- 자판기 프로그램 **VendingMachine**을 다음 조건에 맞게 구현하고 테스트하세요.
  - 음료 종류(5)와 과자 종류(3)를 **라디오 버튼**으로 구현하여 하나만 선택 가능 (종류별 Border 설정)
    - 음료 종류 = 콜라(1000원), 사이다(1000원), 오렌지주스(1800원), 커피(1500원), 물(500원)
    - 과자 종류 = 감자칩(1500원), 초콜릿(1000원), 에너지바(2000원)
  - 수량과 입금액(원)을 입력 받음
    - 입력이 숫자가 아니면 `수량 또는 입금액이 올바르지 않습니다. 다시 입력해 주세요.`를 출력
    - `matches("\\d+")`를 사용하여 입력이 숫자인지 검사
  - 결제 방식(현금, 카드)을 **콤보 박스**로 구현
    - 현금 선택 시, 입금액을 입력 받음(활성화)
    - 카드 선택 시, 입금액 입력 필드를 비우고 비활성화
  - 상품 선택, 수량 입력, 결제 방식 선택 후 **계산** 버튼 클릭 시, **영수증** 출력
    - 상품 선택이 되어 있지 않으면 `상품을 선택해 주세요.`를 출력
    - 정상적으로 계산되어 영수증 출력 시, 모든 선택과 입력은 지워짐 (출력은 지워지지 않음)

## 문제 2 (60점)

- 입력한 입금액이 부족하면 3000원이 부족합니다. 다시 입력해 주세요. 를 출력
- 결과는 텍스트 영역(Text Area)에 출력 (스크롤바 추가하여 구현)
- 영수증은 구매 내역, 총 구매액, 받은 돈, 거스름돈을 포함
  - 구매 내역 = 상품명, 수량, 금액(가격\*수량)

- 형식

===== 영수증 =====		
상품명	수량	금액
초콜릿	5	5000원
=====		
총구매액	5	5000원
받은돈		10000원
거스름돈		5000원

```
"===== 영수증 =====\n"
"상품명\t수량\t금액\n"
"초콜릿\t5\t5000원\n"
"===== \n"
"총구매액\t5\t5000원\n"
"받은돈\t\t10000원\n"
"거스름돈\t\t5000원\n"
```

- 취소 버튼 클릭 시, 모든 선택과 입출력 지워짐 (즉, 초기화면으로 돌아감)
- 텍스트 영역에 출력되는 모든 결과는 취소/종료 버튼 클릭을 제외하고는 계속 남아있어야 함

## 문제 2 (60점)

```
public class VendingMachine extends JFrame {
    private JTextField quantityField, moneyField;
    // TODO : 필요한 필드 자유롭게 선언

    public VendingMachine() {
        setTitle("자판기");
        setSize(500, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        // 제품 패널
        JPanel productPanel = new JPanel(new GridLayout(2, 1, 0, 5));

        JPanel drinkPanel = new JPanel(new GridLayout(0, 3));
        // TODO : 음료 선택 패널 구현

        JPanel snackPanel = new JPanel(new GridLayout(0, 3));
        // TODO : 과자 선택 패널 구현

        productPanel.add(drinkPanel);
        productPanel.add(snackPanel);

        // 수량 및 금액 입력
        JPanel inputPanel = new JPanel();
        inputPanel.add(new JLabel("수량:"));
        quantityField = new JTextField(5);
        inputPanel.add(quantityField);

        inputPanel.add(new JLabel("결제 방식:"));
        // TODO : 결제 방식 콤보박스 구현
        inputPanel.add(/*콤보박스*/);

        inputPanel.add(new JLabel("입금액(원):"));
        moneyField = new JTextField(10);
        inputPanel.add(moneyField);

        // TODO : 결과 출력 영역(TextArea) 스크롤바 추가하여 구현

        // 버튼 패널
        JPanel buttonPanel = new JPanel();
        JButton orderButton = new JButton("계산");
        JButton resetButton = new JButton("취소");
        JButton exitButton = new JButton("종료");

        orderButton.addActionListener(e -> checkOut());
        resetButton.addActionListener(e -> resetForm());
        exitButton.addActionListener(e -> System.exit(0));
    }
}
```

+ 12장 강의 자료 p.7에 있는JCheckBox의 메소드들  
라디오 버튼도 동일하게 사용 가능

```
buttonPanel.add(orderButton);
buttonPanel.add(resetButton);
buttonPanel.add(exitButton);

// 프레임에 추가
JPanel panel = new JPanel(new BorderLayout());
panel.add(productPanel, BorderLayout.CENTER);
panel.add(inputPanel, BorderLayout.SOUTH);
add(panel, BorderLayout.NORTH);
add(/*구현한 컴포넌트*/, BorderLayout.CENTER); // 스크롤바 적용하여 구현한 결과 출력 영역을 프레임에 추가
add(buttonPanel, BorderLayout.SOUTH);
```

```
setVisible(true);
```

```
private void resetForm() {
    // TODO : 초기 화면
}
```

```
public void checkOut() {
    // TODO : 계산 버튼 클릭 시 동작하는 코드
}
```

```
// TODO : 그외 필요한 코드 자유롭게 작성
```

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

## 문제 2 (60점)

- 클래스 명 : **VendingMachine** 📄 VendingMachine.java

실행결과 예시 :

The screenshot shows a Java Swing window titled "자판기" (Vending Machine). The window has a standard title bar with minimize, maximize, and close buttons. The main content area is divided into several sections:

- 음료 선택** (Drink Selection): A group box containing three radio buttons for "콜라(1000원)" (Cola, 1000 won), "사이다(1000원)" (Soda, 1000 won), and "오렌지주스(1800원)" (Orange Juice, 1800 won). Below these are two more radio buttons for "커피(1500원)" (Coffee, 1500 won) and "물(500원)" (Water, 500 won).
- 과자 선택** (Snack Selection): A group box containing three radio buttons for "감자칩(1500원)" (Potato Chips, 1500 won), "초콜릿(1000원)" (Chocolate, 1000 won), and "에너지바(2000원)" (Energy Bar, 2000 won).
- Payment Section:** A row containing a text field for "수량:" (Quantity), a dropdown menu for "결제 방식:" (Payment Method) with "현금" (Cash) selected, and a text field for "입금액(원):" (Amount Paid (won)).
- Buttons:** At the bottom of the window, there are three buttons: "계산" (Calculate), "취소" (Cancel), and "종료" (End).