

# 컴프2 2주차 : 가위바위보 구현

과제 목표 : 클래스를 활용하여 가위바위보 클래스를 구현한다.

학번 : 201902722

이름 : 유형곤

큰 과제 : 가위바위보 구현하기

작은 과제 : Player 클래스 구현하기

문제해결과정 및 코드 설명 :

코드는 다음과 같습니다.

```
private String[] HANDS = {"가위", "바위", "보"};
private int hand;
private String name;
private int point = 0;
```

멤버변수 :

Player 클래스에는 크게 4개의 멤버변수가 있는데,

hand는 가위, 바위, 보를 숫자로 표현한 것이고

HANDS는 hand를 인덱스로 삼아서 가위, 바위, 보를 숫자에서 문자열로 바꾸게 해주는 배열입니다. 그리고 point는 해당 플레이어의 점수입니다.

메서드 :

```
public Player(String name) {  
    this.name = name;  
}  
  
public void setHand(int hand) {  
    this.hand = hand;  
}  
  
public String getHand() {  
    return HANDS[hand];  
}  
  
public void win() {  
    point += 2;  
}  
  
public void lose() {  
    point--;  
}  
  
public void draw() {  
    point++;  
}
```

Player(String name) : Player의 생성자로서, 객체를 생성하고 name을 저장합니다.

setHand(int hand) : hand 변수를 매개 변수로 받은 값으로 수정합니다.

getHand() : hand 변수를 HANDS 배열을 통해서 문자열로 바꾼 후 반환합니다.

win() : 플레이어가 승리했다는 의미로, 점수를 +2 합니다.

lose() : 플레이어가 패배했다는 의미로, 점수를 -1 합니다.

draw() : 플레이어가 비겼다는 의미로, 점수를 +1 합니다.

```

public int getPoint() {
    return point;
}

public String getName() {
    return name;
}

@Override
public String toString() {
    return name + ": " + point;
}

```

getPoint() : 포인트를 반환합니다.

getName() : 이름을 반환합니다.

toString() : Player 객체를 출력할 수 있도록 이름: 포인트의 형태로 반환합니다.

## 작은 과제 2 : Referee 클래스 구현하기

문제해결과정 및 코드 설명 :

멤버변수 :

Referee는 따로 멤버변수가 없습니다.

메서드 :

```

private void givePoint(Player p1, Player p2, int winner) {
    if(winner == 1) {
        p1.win();
        p2.lose();
    } else if(winner == 2) {
        p2.win();
        p1.lose();
    } else {
        p1.draw();
        p2.draw();
    }
}

```

givePoint(Player p1, Player p2, int winner)

승자에게 포인트를 주는 함수로, winner가 1이면 p1이, winner가 2이면 p2가 승자라는 의미이며 winner가 1, 2도 아니면 비겼다고 처리하는 함수입니다.

그래서 각각의 경우(p1이 승자, p2가 승자, 무승부)에서 win, lose, draw 함수를 적절히 호출하여 점수를 조정합니다.

```

private boolean isFirstPlayerWin(String hand1, String hand2) {
    if(hand1.equals("가위") && hand2.equals("보")) {
        return true;
    } else if(hand1.equals("바위") && hand2.equals("가위")) {
        return true;
    } else if(hand1.equals("보") && hand2.equals("바위")) {
        return true;
    } else {
        return false;
    }
}

```

isFirstPlayerWin(String hand1, String hand2) :

p1이 이겼는지를 판단합니다.

P1: 가위, P2: 보

P1, 바위, P2: 가위

P1, 보, P2: 바위

세 가지 경우에 true를, 그 외의 경우(비겼거나 p2가 승리)에 false를 출력합니다.

```

private String getPlayerPoints(Player p1, Player p2) {
    return String.format("%s/ %s", p1.toString(), p2.toString());
}

```

getPlayerPoints(Player p1, Player p2) :

각각의 플레이어의 이름과 점수를 반환합니다.

편의를 위해서 String.format 함수를 사용했습니다.

```

public String getWinner(Player p1, Player p2) {
    boolean didFirstWin = isFirstPlayerWin(p1.getHand(), p1.getHand());
    String msg = null;

    if(didFirstWin) {
        givePoint(p1, p2, 1);
        msg = p1.getName() + "이(가) 이겼습니다.";
    } else {
        if(p1.getHand().equals(p2.getHand())){
            givePoint(p1, p2, 0);
            msg = "비겼습니다.";
        } else {
            givePoint(p1, p2, 2);
            msg = p2.getName() + "이(가) 이겼습니다.";
        }
    }
    return msg + getPlayerPoints(p1, p2);
}

```

getWinner(Player p1, Player p2) :

우선 p1이 승자인지를 판별하고 (isFirstPlayerWin 호출)

True인 경우 p1에게 점수를 줍니다. (givePoint 호출)

False인 경우, p2가 이겼거나 비긴 경우인데,

두 플레이어의 getHand()가 같은 경우 비긴 것으로,

다른 경우에는 p2가 이긴 것으로 판정하여

givePoint를 호출했습니다. 그러고나서 메시지 (이겼습니다 / 비겼습니다) + getPlayerPoint(p1, p2)를 호출합니다.

## 작은 과제 : JUnit으로 가위바위보 테스트하기

JUnit으로 user, rival을 생성한 후 hand를 정한 다음, getWinner의 리턴값을 비교해서 테스트케이스를 작성하였습니다. 테스트 케이스는 총 p1이 이긴 경우, p2가 이긴 경우, 비긴 경우 세 가지로 만들었습니다.

```
@Test
void testgetWinner3() {
    Player user = new Player("User");
    Player rival = new Player("Rival");
    Referee referee = new Referee();

    user.setHand(1);
    rival.setHand(1);

    assertEquals(referee.getWinner(user, rival) , "비겼습니다.(User: 1/ Rival: 1)");
}

,

@Test
void testgetWinner2() {
    Player user = new Player("User");
    Player rival = new Player("Rival");
    Referee referee = new Referee();

    user.setHand(1);
    rival.setHand(0);

    assertEquals(referee.getWinner(user, rival) , "User이(가) 이겼습니다.(User: 2/ Rival: -1)");
}

@Test
void testgetWinner1() {
    Player user = new Player("User");
    Player rival = new Player("Rival");
    Referee referee = new Referee();

    user.setHand(0);
    rival.setHand(1);

    assertEquals(referee.getWinner(user, rival) , "Rival이(가) 이겼습니다.(User: -1/ Rival: 2)");
}
```