

REPORT



- 과목명 : 객체지향프로그래밍
- 담당교수 : 엄진영 교수님
- 학과 : 컴퓨터공학과
- 학번 : 2019112130
- 이름 : 조 양 진

객체지향프로그래밍 12주차 실습과제 보고서

2019112130 조양진

문제 1.

Blackjack

- ✓ 11주차에 진행했던 과제 규칙을 이용하여 진행
- ✓ 딜러와 플레이어는 하나의 클래스를 사용하며 이 클래스 안에 해당 플레이어의 블랙잭 확률을 구하는 함수를 정적 함수로 만든다.
- ✓ 게임에 사용되는 카드는 (String 배열) 정적 변수를 사용한다. (이때 카드 무늬는 무시한다.)

- 문제 분석

11주차 과제를 기반으로, 주어진 조건에 맞춰 블랙잭 게임 코드를 수정해야 합니다. 우선 딜러와 플레이어가 하나의 클래스를 사용하여야 하며, 이 클래스 안에 해당 플레이어의 블랙잭 확률을 구하는 함수를 정적으로 구현해야 합니다. 또한 게임에 사용되는 카드 (덱)은 String형의 배열을 정적으로 사용해야 합니다.

저는 11주차에 한 class안에 모든 것을 구현했었기 때문에, 이를 새로운 클래스에 이식하는 작업이 필요합니다.

또한 기존의 코드에서는 덱의 카드를 int형 배열로 사용했기 때문에, 이를 String 형으로 바꾸고 계산을 Integer.parseInt나 Integer.toString 과 같은 함수를 사용해야 합니다.

그리고 주어진 조건 1. 딜러와 플레이어가 공통으로 사용하는 하나의 클래스 안에 블랙잭 확률 (21점을 만들 확률)을 정적 (static) 함수로 만들고 2. 게임에 사용되는 카드 (덱)을 String 정적 배열 변수로 바꾸는 것을 우선 순위로 했습니다.

- 코드 설명 (week12 - Blackjack.java, Player.java)

11주차에 제출했던 코드와는 조금 다르게 우선 Player 라는 클래스를 만들고 이 Player 클래스를 사용하여 컴퓨터 (CPU / Dealer)와 사용자 (User / Player)가 같이 사용하도록 해야 했습니다.

```
public class Player
```

```
public int turn – 플레이어의 턴 수
```

public static String[] deck – 게임에 사용되는 덱 (조건 1)
 public int cardPoint – 해당 플레이어의 합산 점수
 public int[] cardSet – 해당 플레이어의 카드 패
 public void drawCard() – 카드 뽑기 함수
 public void printCard() – 해당 플레이어가 들고 있는 카드 출력 함수
 public static void analyze(int point) – 해당 플레이어가 블랙잭을 만들 수 있는 확률을
 구해주는 정적 함수 (조건 2)

public class Blackjack

public static Player Cpu = new Player() – CPU/dealer 객체 생성
 public static Player User = new Player() – User/player 객체 생성

public static void main(String[] args) – 메인 함수
 public static int calculate(int[] input, int turnCounter) – 카드 점수를 가장 유리하게
 계산해주는 함수
 public static void normalEndGame() – 게임이 종료될 때 실행되는 함수

- 결과 및 분석

```

1 package week11;
2
3 // 2019112130 조양현
4
5 package week11;
6
7 // 딜러와 플레이어 2명이 존재한다
8 // 카드는 52장을 포함한다
9 // 2-10은 숫자, J, Q, K는 10으로, A는 1 또는 11, 조지는 가장
10 // 게임 시작과 동시에 딜러와 플레이어는 순차적으로 카드를 하나씩 뽑아 차지
11 // 딜러가 게임을 진행하며 딜러는 카드를 합계 점수가 16점 이하이면 반드시
12 // 플레이어는 딜러가 카드를 추가할 것을 기다린다
13 // 만약 21 이하인 경우 딜러와 플레이어 중 소유했던 카드의 합이 21에 가까운
14 // 카드를 가진 쪽이 승리한다
15 // 딜러와 플레이어의 카드는 전부 요인되어 있으며 플레이어는 블랙잭의 확률
16
17 import java.util.Scanner;
18
19 public class Blackjack {
20
21     // CPU 객체 생성 - Dealer
22     public static Player Cpu = new Player();
23     // User 객체 생성 - Player
24     public static Player User = new Player();
25     static boolean endGame = false; // 게임 반복 플래그
26
27     @SuppressWarnings("static-access")
28     public static void main(String[] args) {
29         Scanner sc = new Scanner(System.in);
30         int command = 0; // 사용자 입력 번호
31
32         System.out.println("Blackjack Game");
33
34         System.out.println("딜러가 첫 두장의 카드를 뽑습니다.");
35         Cpu.drawCard();
36         Cpu.printCard();
37         // 뽑은 카드 출력 함수
38         System.out.println("플레이어가 첫 두장의 카드를 뽑습니다.");
39         User.drawCard();
40         User.printCard();
41         // 뽑은 카드 출력 함수
42         User.printCard();
43
44         while(true) {
45             // 반복되는 게임
46             Cpu.cardPoint = calculate(Cpu.cardSet, Cpu.turn);
47             User.cardPoint = calculate(User.cardSet, User.turn);
48             System.out.print("\n점수를 입력해주세요. (1)Hit, (2)Stay, (3)
49             try {
50                 command = sc.nextInt();
51                 System.out.println();
52             } catch (Exception e) {}
53             if(command == 1) {
54                 // 게임 종료 함수
55                 normalEndGame();
56             }
57         }
58     }
59
60     // 딜러와 플레이어는 하나의 클래스를 사용하여 이
61     // 클래스 안에 해당 플레이어의 블랙잭 확률을 구하는
62     // 함수를 정의 함수로 만든다
63
64     // 게임에 사용되는 카드는 (String 배열) 정적 변수를
65     // 사용한다 이때 카드는 무한하다
66
67     import java.util.Random;
68     public class Player {
69         public int turn = 0;
70
71         // 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
72         public static String[] deck = {"2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A"};
73         public int cardPoint = 0;
74         public int[] cardSet = new int[52];
75
76         // 랜덤 변수 생성
77         public void drawCard() {
78             Random rand = new Random();
79             int temp = rand.nextInt(52);
80             // cardSet에서 1 번고 뒤로
81             if(Integer.parseInt(deck[temp]) == 0) {
82                 // 해당 숫자의 카드가 없다면 (11 번)
83                 drawCard();
84             } else {
85                 deck[temp] = Integer.parseInt(Integer.parseInt(deck[temp]) + 1);
86                 cardSet[turn] = temp;
87                 turn++;
88             }
89         }
90
91         public void printCard() {
92             for(int i=0; i<turn; i++) {
93                 if(cardSet[i] >= 10) {
94                     System.out.print("A ");
95                     cardSet[i] = 1;
96                 } else if(cardSet[i] == 9) {
97                     System.out.print("J ");
98                 } else if(cardSet[i] == 8) {
99                     System.out.print("Q ");
100                } else if(cardSet[i] == 7) {
101                    System.out.print("K ");
102                } else if(cardSet[i] == 6) {
103                    System.out.print(" ");
104                } else if(cardSet[i] == 5) {
105                    System.out.print(" ");
106                } else if(cardSet[i] == 4) {
107                    System.out.print(" ");
108                } else if(cardSet[i] == 3) {
109                    System.out.print(" ");
110                } else if(cardSet[i] == 2) {
111                    System.out.print(" ");
112                }
113            }
114        }
115    }
116
117     // 게임이 종료될 때 실행되는 함수
118     public static void normalEndGame() {
119         // 게임 종료 함수
120         System.out.println("게임이 종료되었습니다.");
121         System.out.println("플레이어 점수: " + User.cardPoint);
122         System.out.println("딜러 점수: " + Cpu.cardPoint);
123         System.out.println("게임이 종료되었습니다.");
124     }
125
126     // 카드 점수를 가장 유리하게 계산해주는 함수
127     public static int calculate(int[] input, int turnCounter) {
128         // 카드 점수를 가장 유리하게 계산해주는 함수
129         int sum = 0;
130         for(int i=0; i<input.length; i++) {
131             sum += input[i];
132         }
133         return sum;
134     }
135 }
    
```

- eclipse 상에서 실행한 모습

```
Console [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (2020. 1
Blackjack Game
딜러가 첫 두장의 카드를 뽑습니다.
A J
플레이어가 첫 두장의 카드를 뽑습니다.
Q J

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 3

49장 중 4장남은 A 또는 Joker를 뽑아 21을 만들 확률은 8.16%입니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 4

더이상 Dealer는 카드를 뽑지 않습니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 1

Dealer:
A J
Player:
Q J 8

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 3

Over 21, BUSTED

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 4

더이상 Dealer는 카드를 뽑지 않습니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 2

Dealer Points: 21
Player Points: 28
BUST: Dealer WON
```

- 게임 테스트 스크린샷 #1

```
Console [X]
<terminated> Blackjack (1) [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (2020
Blackjack Game
딜러가 첫 두장의 카드를 뽑습니다.
Joker 5
플레이어가 첫 두장의 카드를 뽑습니다.
3 7

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 3

49장 중 4장남은 A 또는 Joker를 뽑아 21을 만들 확률은 8.16%입니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 4

49장 중 3장남은 Joker, 5를 뽑아 21을 만들 확률은 6.12%입니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 1

Dealer:
Joker 5 3
Player:
3 7 A

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 3

Blackjack - 21

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 2
|
Dealer Points: 19
Player Points: 21
Player WON
```

- 게임 테스트 스크린샷 #2

```
Console [X]
<terminated> Blackjack (1) [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (2020
Blackjack Game
딜러가 첫 두장의 카드를 뽑습니다.
10 A
플레이어가 첫 두장의 카드를 뽑습니다.
9 5

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 3

49장 중 5장남은 Joker, 7를 뽑아 21을 만들 확률은 10.20%입니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 4

더이상 Dealer는 카드를 뽑지 않습니다.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 1

Dealer:
10 A
Player:
9 5 10

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 3

Over 21, BUSTED

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player (4) Analyze Dealer: 2
|
Dealer Points: 21
Player Points: 24
BUST: Dealer WON
```

- 게임 테스트 스크린샷 #3

- 소감

이번 12주차 과제는 저번주의 과제의 연장선이라 그렇게 어려움을 느끼진 않았습니다. 다만 제 알고리즘이 general 함과는 조금 동떨어져 있는 것 같아 String 정적 배열로 짜는 것이 큰 의미는 없었을지는 몰라도 조건을 맞추기 위해 최선을 다 했습니다. 또한 11주차 과제 결과물의 아쉬움 중 하나는 제가 프로그램을 절차지향적으로 작성했다는 것이었는데, 이번 12주차 과제를 하면서 플레이어 클래스를 따로 하나 만들면서 조금 더 객체지향 적인 프로그램이 된 것 같아 기쁩니다.

- 부록 (코드 전문)

// 2019112130 조양진

// Blackjack.java

package week12;

/*

* 딜러와 플레이어 2명이 존재한다

* 카드는 조커를 포함한 53장이다

* 2~10은 숫자 그대로 점수를, K/Q/J는 10점으로, A는 1 또는 11, 조커는 가장 유리한 경우로 계산

* 게임 시작과 동시에 딜러와 게이머는 순차적으로 카드를 하나씩 뽑아 각자 2개의 카드를 소지한다

* 딜러부터 게임을 진행하며 딜러는 2카드의 합계 점수가 16점 이하이면 반드시 1장을 추가로 뽑고, 17점 이상이면 STAY한다

* 플레이어는 얼마든지 카드를 추가로 뽑을 수 있다

* 양쪽 다 STAY한 경우 딜러와 플레이어 중 소유한 카드의 합이 21에 가까운 쪽이 승리한다

* 단 21을 초과하면 초과한 쪽이 진다.

* 딜러와 플레이어의 카드는 전부 오픈되어 있으며 플레이어는 블랙잭의 확률을 계산할 수 있어야 한다.

*/

import java.util.Scanner;

public class Blackjack {

// Cpu 객체 생성 - Dealer

public static Player Cpu = **new** Player();

// User 객체 생성 - Player

public static Player User = **new** Player();

static boolean endGame = **false**; // 게임 반복 플래그

@SuppressWarnings("static-access")

public static void main(String[] args) {

Scanner sc = **new** Scanner(System.in);

int command = 0; // 사용자 입력 변수

System.out.println("Blackjack Game");

System.out.println("딜러가 첫 두장의 카드를 뽑습니다.");

Cpu.drawCard();

```

    Cpu.drawCard();
    // 뽑은 카드 출력 함수
    Cpu.printCard();
    System.out.println("플레이어가 첫 두장의 카드를 뽑습니다.");
    User.drawCard();
    User.drawCard();
    // 뽑은 카드 출력 함수
    User.printCard();

    while(!endGame) {
        // 반복되는 입력
        Cpu.cardPoint = calculate(Cpu.cardSet, Cpu.turn);
        User.cardPoint = calculate(User.cardSet, User.turn);
        System.out.print("\n행동을 입력해주세요.\n(1)Hit, (2)Stay,
(3)Analyze Player (4) Analyze Dealer: ");
        try {
            command = sc.nextInt();
            System.out.println();

            if(command == 2) {
                // 게임 종료 함수
                normalEndGame();
            }
            else if(command == 1){
                // 딜러 합 16 체크 - 작으면 딜러 카드 뽑는 함수
                if(Cpu.cardPoint <= 16 && Cpu.turn==2) {
                    Cpu.drawCard();
                }else if(Cpu.cardPoint < 17) {
                    Cpu.drawCard();
                }
                // 플레이어 합 21 오버 체크 - 크면 게임 종료 함수
                if(User.cardPoint > 21) {
                    // 게임 종료 함수
                    normalEndGame();
                    break;
                }
                // 플레이어 합 21 오버 체크 - 크면 게임 종료 함수
                if(Cpu.cardPoint > 21) {
                    // 게임 종료 함수
                    normalEndGame();

```



```

        break;
    }
    // 플레이어 드로우
    User.drawCard();
    // 뽑은 카드 출력 함수
    System.out.println("Dealer: ");
    Cpu.printCard();
    System.out.println("Player: ");
    User.printCard();
}
else if(command == 3) {
    User.analyze(User.cardPoint);
}
else if(command == 4) {
    if(Cpu.cardPoint <= 16 && Cpu.turn==2) {
        Cpu.analyze(Cpu.cardPoint);
    }else if(Cpu.cardPoint < 17) {
        Cpu.analyze(Cpu.cardPoint);
    }else {
        System.out.println("더이상 Dealer는 카드를
뽑지 않습니다.");
    }
}
}
} catch (Exception e) {
    System.out.println("잘못된 입력입니다. 다시 입력해주세요.");
    e.printStackTrace();
    throw e;
}
}
}

```

```

// 카드 점수 계산 알고리즘 함수
public static int calculate(int[] input, int turnCounter) {
    int total = 0;
    int aCounter = 0;
    boolean isJokerIn = false;
    // 1. 2~10,J,Q,K는 우선 처리
    for(int i=0; i<turnCounter; i++) {
        if(input[i] == 0) {
            total +=2;

```

```

    }else if(input[i]==1){
        total +=3;
    }else if(input[i]==2){
        total +=4;
    }else if(input[i]==3){
        total +=5;
    }else if(input[i]==4){
        total +=6;
    }else if(input[i]==5){
        total +=7;
    }else if(input[i]==6){
        total +=8;
    }else if(input[i]==7){
        total +=9;
    }else if(input[i]==8){
        total +=10;
    }else if(input[i]==9){
        total +=10;
    }else if(input[i]==10){
        total +=10;
    }else if(input[i]==11){
        total +=10;
    }else if(input[i]==12) {
        aCounter++;
    }else if(input[i]==13) {
        isJokerIn = true;
    }
}

// 2. A의 개수를 저장해놨다가 1번과 비교하여 더하기
if(aCounter > 0) {
    for(int j=0; j<aCounter; j++) {
        if(total < 11) {
            total += 11;
        }else {
            total += 1;
        }
    }
}

// 3. 조커의 유무
if(isJokerIn) {

```

```

        // total값이 1과 11 사이라면 total 21에 맞출 수 있도록 더합니다
        if((21-total)>=1 && (21-total)<=11) {
            total += (21-total);
            // total값이 11보다 작다면 11을 더합니다
        }else if(total<11) {
            total += 11;
            // 나머지 경우는 1을 더합니다
        }else {
            total +=1;
        }
    }
    // total을 반환합니다
    return total;
}

```

// 게임 종료 함수

```

public static void normalEndGame() {
    System.out.println("Dealer Points: " + Cpu.cardPoint);
    System.out.println("Player Points: " + User.cardPoint);
    endGame = true;
    if(User.cardPoint > 21) {
        System.out.println("BUST: Dealer WON");
    }else if(Cpu.cardPoint > 21) {
        System.out.println("BUST: Player WON");
    }else if(User.cardPoint == Cpu.cardPoint) {
        System.out.println("DRAW");
    }else if(User.cardPoint-Cpu.cardPoint>0) {
        System.out.println("Player WON");
    }else if(User.cardPoint-Cpu.cardPoint<0) {
        System.out.println("Dealer WON");
    }
}

```

}

// 2019112130 조양진

// Player.java

package week12;

```
/*
딜러와 플레이어는 하나의 클래스를 사용하며 이
클래스 안에 해당 플레이어의 블랙잭 확률을 구하는
함수를 정적 함수로 만든다
```

```
게임에 사용되는 카드는 (String 배열 ) 정적 변수를
사용한다 이때 카드 무늬는 무시한다
```

```
*/
```

```
import java.util.Random;
```

```
public class Player {
```

```
    // 플레이어의 턴 수
```

```
    public int turn = 0;
```

```
    // 2, 3, 4, 5, 6, 7, 8, 9, 10, K,
```

```
    Q, J, A, Joker
```

```
    public static String[] deck = {"4","4","4","4","4","4","4","4","4","4","4","4","4","1"};
```

```
    // 카드 합산 점수
```

```
    public int cardPoint = 0;
```

```
    // 플레이어의 카드 패
```

```
    public int[] cardSet = new int[60];
```

```
    // 카드 뽑기 함수
```

```
    public void drawCard() {
```

```
        Random rand = new Random();
```

```
        // 0~13
```

```
        int temp = rand.nextInt(14);
```

```
        // cardSet에서 1 빼고 비교
```

```
        if((Integer.parseInt(deck[temp]) == 0)) {
```

```
            // 해당 숫자의 카드가 없다면 다시 뽑기
```

```
            drawCard();
```

```
        }else {
```

```
            deck[temp] = Integer.toString(Integer.parseInt(deck[temp]) - 1);
```

```
            cardSet[turn] = temp;
```

```
            turn++;
```

```
        }
```

```
    }
```

```
    // 들고 있는 카드 출력 함수
```

```
    public void printCard() {
```

```
        for(int i=0; i<turn; i++) {
```

```
            if(cardSet[i] >=0 && cardSet[i] <=8) {
```

```

        System.out.printf(" %d ", cardSet[i]+2);
    }
    else if(cardSet[i]==9) {
        System.out.print(" K ");
    }
    else if(cardSet[i]==10) {
        System.out.print(" Q ");
    }
    else if(cardSet[i]==11) {
        System.out.print(" J ");
    }
    else if(cardSet[i]==12) {
        System.out.print(" A ");
    }
    else if(cardSet[i]==13) {
        System.out.print(" Joker ");
    }
}
System.out.println();
}

```

// 해당 플레이어의 블랙잭 확률을 구하는 함수를 정적 함수로

```

public static void analyze(int point) {
    // 21점을 만들 수 있는 확률을 계산하는 조건문
    if(point < 21) {
        int temp = 21 - point;
        if(temp > 11) {
            System.out.print("21을 만들려면 한 장 더
뽑아야 합니다.");
        }else {
            int total = 0;
            int winTemp = 0;
            float winCase = 0;
            for(int i=0; i<14; i++) {
                total += Integer.parseInt(deck[i]);
            }
            if((temp == 11) || (temp == 1)) {
                winTemp = Integer.parseInt(deck[12])
+ Integer.parseInt(deck[13]);

                winCase = (float)winTemp/total;
            }
        }
    }
}

```

```

        System.out.printf("%d장 중 %d장남은
A 또는 Joker를 뽑아 21을 만들 확률은 %.2f%%입니다.", total, winTemp, winCase*100);
    }else if(temp == 10) {
        winTemp = Integer.parseInt(deck[8]) +
Integer.parseInt(deck[9]) + Integer.parseInt(deck[10]) + Integer.parseInt(deck[11]) +
Integer.parseInt(deck[13]);

        winCase = (float)winTemp/total;
        System.out.printf("%d장 중 Joker, 10, K, Q
또는 J를 %d장을 뽑아 21을 만들 확률은 %.2f%%입니다.", total, winTemp, winCase*100);
    }else if(temp == 9) {
        winTemp = Integer.parseInt(deck[7]) +
Integer.parseInt(deck[13]);

        winCase = (float)winTemp/total;
        System.out.printf("%d장 중 %d장남은
Joker, 9를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
    }else if(temp == 8) {
        winTemp = Integer.parseInt(deck[6]) +
Integer.parseInt(deck[13]);

        winCase = (float)winTemp/total;
        System.out.printf("%d장 중 %d장남은
Joker, 8를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
    }else if(temp == 7) {
        winTemp = Integer.parseInt(deck[5]) +
Integer.parseInt(deck[13]);

        winCase = (float)winTemp/total;
        System.out.printf("%d장 중 %d장남은
Joker, 7를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
    }else if(temp == 6) {
        winTemp = Integer.parseInt(deck[4]) +
Integer.parseInt(deck[13]);

        winCase = (float)winTemp/total;
        System.out.printf("%d장 중 %d장남은
Joker, 6를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
    }else if(temp == 5) {
        winTemp = Integer.parseInt(deck[3]) +
Integer.parseInt(deck[13]);

        winCase = (float)winTemp/total;
        System.out.printf("%d장 중 %d장남은
Joker, 5를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
    }else if(temp == 4) {

```

```

Integer.parseInt(deck[13]);

winTemp = Integer.parseInt(deck[2]) +

winCase = (float)winTemp/total;
System.out.printf("%d장 중 %d장남은
Joker, 4를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
}else if(temp == 3) {
winTemp = Integer.parseInt(deck[1]) +

Integer.parseInt(deck[13]);

winCase = (float)winTemp/total;
System.out.printf("%d장 중 %d장남은
Joker, 3를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
}else if(temp == 2) {
winTemp = Integer.parseInt(deck[0]) +

Integer.parseInt(deck[13]);

winCase = (float)winTemp/total;
System.out.printf("%d장 중 %d장남은
Joker, 2를 뽑아 21을 만들 확률은 %.2f%%입니다.", total,winTemp, winCase*100);
}else if (temp == 0) {
System.out.println("이미 21점입니다.");
}
System.out.println();
}
}else if (point > 21) {
System.out.println("Over 21, BUSTED");
}else {
System.out.println("Blackjack - 21");
}
}
}
}

```