

REPORT



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

객체지향프로그래밍 13주차 실습과제 보고서

2019112130 조양진

문제 1.

Blackjack

- ✓ 11주차에 진행했던 과제 규칙을 이용하여 진행
- ✓ Try - Catch 문을 사용해서 플레이어가 카드를 받을지 말지 결정하는 단계에서 그 외의 선택지를 고를 경우 에러를 발생시켜 Catch 문에서 경고문을 작성하고 제대로 된 명령이 들어올 때까지 반복하세요.

- 문제 분석

11, 12주차에 이어서 같은 블랙잭 프로그램을 사용하여 이번에는 try-catch로 exception 처리를 해보고 wav 음원 파일을 재생시켜 보는 문제입니다. 우선 첫번째 조건은 플레이어가 카드를 받을 지 말지 결정하는 단계에서 - try를 사용하고 예상치 못한 input에 대해 catch를 하는 것으로 보입니다. 또한 Throw new Exception()을 사용하여 에러를 발생 시켜야 합니다. 두번째 조건은 1. 플레이어가 카드가 받겠다고 하는 경우, 2. 플레이어가 이겼을 때, 3. 플레이어가 졌을 때, 4. 게임 시작시에 주어진 음원을 사용하여 소리를 재생시켜야 합니다. 이는 주어진 코드를 조금 변형하여 사용하면 될 것 같습니다.

- 코드 설명 (week13 - Blackjack.java, Player.java, MediaPlayer.java)

12주차의 코드를 이어서 수정했습니다. 이번에는 MediaPlayer라는 클래스를 새로 생성하여 2번째 조건인 프로그램에서 필요한 때에 음원을 재생할 때 사용하는 객체를 정의했습니다. 우선 첫번째 조건인 try-catch에 대해 설명드리겠습니다. 이는 main 함수가 정의되어 있는 Blackjack.java 파일에 작성했습니다.

```

@SuppressWarnings("static-access")
public static void main(String[] args) throws Exception {
    Scanner sc = new Scanner(System.in);
    int command = 0; // 사용자 입력 변수

    System.out.println("Blackjack Game");
    MediaPlayer gamestart = new MediaPlayer();
    gamestart.play("game_start");

    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)
        |
        try {
            sc.reset();
            command = 0;
            command = sc.nextInt();
            System.out.println();

```

main 함수 안에서 처음 게임이 실행되고 난 후 딜러의 두장, 플레이어의 두장을 뽑은 이후 while 반복문 안에 try 구문을 집어넣었습니다. try 이전에 입력할 수 있는 옵션을 제공하여 try-catch를 해도 메뉴가 다시 출력되도록 하였습니다. 이후 try 밑의 구문은 거의 동일하고

```

        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.");
        }
    }
    else {
        throw new Exception("\n잘못된 입력입니다. 다시 입력해주세요.");
    }
} catch (Exception e) {
    System.out.println("\n잘못된 입력입니다. 다시 입력해주세요.");
    sc = new Scanner(System.in);
}
}
}

```

제공된 옵션을 사용하지 않을 경우 Exception을 throw 하도록 설정하여 catch 받을 수 있도록 하였습니다. 따라서 제가 제시한 옵션 (1, 2, 3, 4) 외의 옵션이 입력되어도 catch 받아 다시 메뉴를 입력받을 수 있게 하였습니다. 결과 예시는 결과 및 분석에서 설명 드리겠습니다.

이후 두번째 조건이었던 음원 재생은 MediaPlayer.java에서 처리하였습니다.

```

package week13;

import java.io.File;
import javax.sound.sampled.*;

public class MediaPlayer {
    // 인자로 파일 이름명을 받음 (확장자 제외)
    public void play(String name) {
        File bgm;
        AudioInputStream stream;
        AudioFormat format;
        DataLine.Info info;

        // 음원의 절대 경로 - 다른 컴퓨터에선 변경 필요
        bgm = new File("C:\\Users\\70339\\OneDrive\\"
            + "\\바탕 화면\\Github\\Workspace\\week13\\src\\week13\\"+name+".wav");

        Clip clip;
        try {
            stream = AudioSystem.getAudioInputStream(bgm);
            format = stream.getFormat();
            info = new DataLine.Info(Clip.class, format);
            clip = (Clip)AudioSystem.getLine(info);
            clip.open(stream);
            clip.start();
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }
}

```

조교님께서 주셨던 예시 소스코드를 참고하여 조금 변형 시켜서 적용하였습니다. 이 MediaPlayer 클래스는 play라는 함수를 가지고 있으며, play 함수 인자로 string이 들어가는데, 이 string에 음원 파일 명을 집어넣으면 해당 음원 (wav파일)을 실행합니다.

bgm 변수의 경우는 음원의 절대 경로를 설정했습니다. 따라서 다른 PC에서는 음원이 존재하는 경로로 바꾸어줘야 합니다.

이 MediaPlayer 클래스를 프로그램에 4 군데에 적용해야 하는데,

1. 게임이 시작할 때

```
@SuppressWarnings("static-access")
public static void main(String[] args) throws Exception {
    Scanner sc = new Scanner(System.in);
    int command = 0; // 사용자 입력 변수

    System.out.println("Blackjack Game");
    MediaPlayer gamestart = new MediaPlayer();
    gamestart.play("game_start");
```

Blackjack.java의 main 함수 안에 게임이 시작할 때 game_start 음원을 재생하도록 하였습니다.

2. 카드를 뽑을 때

```
}
else if(command == 1){
    MediaPlayer hit = new MediaPlayer();
    hit.play("hit_the_card");
```

마찬가지로 Blackjack.java의 main 함수의 while 반복 문 안의 if-else if 문에서 command에 1이 들어 왔을 때 (hit 명령) hit_the_card 음원을 재생하도록 하였습니다.

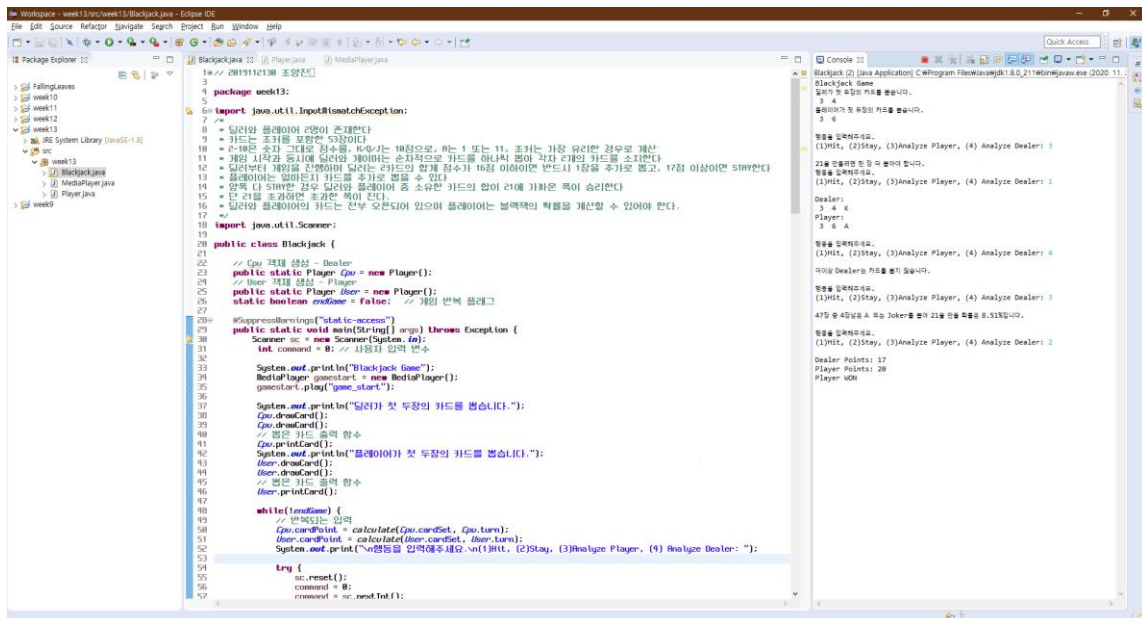
3. 플레이어가 졌을 때
4. 플레이어가 이겼을 때

// 게임 종료 함수

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

제 블랙잭 게임에서 모든 게임의 끝은 Blackjack.java 코드 안의 normalEndGame 함수를 거칩니다. 이 함수 안에서 딜러가 이겼을 때는 lose.wav, 플레이어가 이겼을 때는 win.wav 파일을 재생합니다.

- 결과 및 분석



- eclipse 상에서 실행한 모습

```
Console
<terminated> Blackjack (2) [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (2020. 11. 28. 오전 1:26:07)
Blackjack Game
딜러가 첫 두장의 카드를 뽑습니다.
2 10
플레이어가 첫 두장의 카드를 뽑습니다.
A Q

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player, (4) Analyze Dealer: 5

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

49장 중 5장남은 Joker, 9를 뽑아 21을 만들 확률은 10.20%입니다.

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

Dealer Points: 12
Player Points: 21
Player WON
```

- 게임 테스트 스크린샷 #1

```

Blackjack Game
딜러가 첫 두장의 카드를 뽑습니다.
3 4
플레이어가 첫 두장의 카드를 뽑습니다.
6 2

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

Dealer:
3 4 Q
Player:
6 2 4

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

47장 중 5장남은 Joker, 9를 뽑아 21을 만들 확률은 10.64%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:
3 4 Q
Player:
6 2 4 5

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

Dealer:
3 4 Q
Player:
6 2 4 5 A

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

45장 중 4장남은 Joker, 3를 뽑아 21을 만들 확률은 8.89%입니다.

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

Dealer Points: 17
Player Points: 18
Player WON

```

- 게임 테스트 스크린샷 #2


```
Console
Blackjack (2) [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (2020. 11. 28. 오전 2:04:28)
Blackjack Game
딜러가 첫 두장의 카드를 뽑습니다.
8 5
플레이어가 첫 두장의 카드를 뽑습니다.
7 J

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

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

잘못된 입력입니다. 다시 입력해주세요.

행동을 입력해주세요.
(1)Hit, (2)Stay, (3)Analyze Player, (4) Analyze Dealer: 1q2w3e4r5t

잘못된 입력입니다. 다시 입력해주세요.

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

Dealer Points: 13
Player Points: 17
Player WON
```

- 게임 테스트 스크린샷 #3

영어, 숫자 (1,2,3,4를 제외한), 예상치 못한 입력값을 입력 받았을 때, 잘못된 입력입니다. 다시 입력해주세요. 문구를 출력하고 다시 입력값을 받습니다.

또한 음원이 정상적으로 모두 잘 작동함을 확인했습니다.

- 소감

드디어 3주간 이어져온 블랙잭 게임 과제가 끝이 났습니다. 특히 이번주에 구현한 try-catch문은 정말 중요하여 다른 프로그램에서도 아주 많이 사용되는 것으로 알고 있고 실제로도 저도 많이 써왔습니다. 또한 음원을 재생하는 방법도 알아보았는데, 제가 사용한 방법은 사실 컴퓨터 로컬 저장소에 있는 절대 경로를 지정하여 재생한 것이지만 다음번에 사용할 때에는 프로젝트 안에 포함시켜서 어떤 실행환경에서도 똑 같은 결과값을 얻을 수 있는 방법을 연구해 봐야 할 것 같습니다.

- 부록 (코드 전문)

```
// 2019112130 조양진
```

```
// Blackjack.java
```

```
package week13;
```

```
/*
```

```
 * 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) throws Exception {
    Scanner sc = new Scanner(System.in);
    int command = 0; // 사용자 입력 변수

    System.out.println("Blackjack Game");
    MediaPlayer gamestart = new MediaPlayer();
    gamestart.play("game_start");

    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 {
            sc.reset();
            command = 0;
            command = sc.nextInt();
            System.out.println();

```

방지

```
if(command == 2) {  
    // 게임 종료 함수  
    normalEndGame();  
    // 프로그램이 종료되면 사운드가 씹히는 버그를  
  
    Thread.sleep(2000);  
}
```

함수

```
else if(command == 1){  
    MediaPlayer hit = new MediaPlayer();  
    hit.play("hit_the_card");  
  
    // 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
```

버그를 방지

```
if(User.cardPoint > 21) {  
    // 게임 종료 함수  
    normalEndGame();  
    // 프로그램이 종료되면 사운드가 씹히는  
  
    Thread.sleep(2000);  
    break;  
}  
// 플레이어 합 21 오버 체크 - 크면 게임 종료
```

함수

```
if(Cpu.cardPoint > 21) {  
    // 게임 종료 함수  
    normalEndGame();  
    // 프로그램이 종료되면 사운드가 씹히는  
  
    Thread.sleep(2000);  
    break;
```

버그를 방지

```

    }
    // 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.");
    }
}
else {
    throw new Exception("\n잘못된 입력입니다.
다시 입력해주세요.");
}
} catch (Exception e) {
    System.out.println("\n잘못된 입력입니다. 다시
입력해주세요.");
    sc = new Scanner(System.in);
}
}
}

// 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);
    MediaPlayer sound = new MediaPlayer();
    endGame = true;
    if(User.cardPoint > 21) {
        sound.play("lose");
    }
}

```

```

        System.out.println("BUST: Dealer WON");
    }else if(Cpu.cardPoint > 21) {
        sound.play("win");
        System.out.println("BUST: Player WON");
    }else if(User.cardPoint == Cpu.cardPoint) {
        System.out.println("DRAW");
    }else if(User.cardPoint-Cpu.cardPoint>0) {
        sound.play("win");
        System.out.println("Player WON");
    }else if(User.cardPoint-Cpu.cardPoint<0) {
        sound.play("lose");
        System.out.println("Dealer WON");
    }
}
}
}

```

```

// 2019112130 조양진
// Player.java
package week13;

```

```

/*
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) {
            winTemp =
Integer.parseInt(deck[2]) + Integer.parseInt(deck[13]);
            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");
}
}
}

```

```

}
// 2019112130 조양진
// MediaPlayer.java
package week13;

import java.io.File;
import javax.sound.sampled.*;

public class MediaPlayer {
    // 인자로 파일 이름을 받음 (확장자 제외)
    public void play(String name) {
        File bgm;
        AudioInputStream stream;
        AudioFormat format;
        DataLine.Info info;

        // 음원의 절대 경로 - 다른 컴퓨터에선 변경 필요
        bgm = new File("C:\\Users\\WW70339\\OneDrive\\WW"
            + "바탕
화면\\Github\\Workspace\\week13\\src\\week13\\WW" + name + ".wav");

        Clip clip;
        try {
            stream = AudioSystem.getAudioInputStream(bgm);
            format = stream.getFormat();
            info = new DataLine.Info(Clip.class, format);
            clip = (Clip)AudioSystem.getLine(info);
            clip.open(stream);
            clip.start();
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }
}

```