



# REPORT

Java BlackJack 게임 구현



교 과 목	객체지향프로그래밍
담 당 교 수	이계성 교수님
학 과	소프트웨어학과
학 번	32141868
이 름	박유현
제 출 일	2018.06.01



# 목 차

1. 개요.....	2
2. 블랙잭의 상세 규칙 .....	3
2.1 카드 점수의 계산.....	3
2.2 게임 방법.....	3
2.3 플레이어가 취할 수 있는 행동 .....	3
2.4 게임 결과.....	3
3. 블랙잭 게임 진행 순서도.....	4
4. 클래스의 구현.....	5
4.1 카드와 카드덱의 구현.....	5
4.2 플레이어의 구현.....	6
4.3 블랙잭 AI 플레이어 행동 패턴 구현.....	7
4.3.1 AI 플레이어 행동 패턴 알고리즘.....	7
4.4 UI 를 출력하는 클래스의 구현 .....	8
5. 결과.....	9
5.1 게임 처음 실행 시 나타나는 스플래시 화면.....	9
5.2 플레이어 이름을 설정하는 화면 .....	9
5.3 메인 메뉴.....	10
5.4 게임 방법 설명화면 .....	10
5.5 게임 시작 시 AI 플레이어 추가 화면.....	11
5.6 게임 시작 시 베팅화면 .....	11
5.7 게임 화면 – AI 플레이어 턴.....	12
5.8 게임 화면 – 플레이어 턴.....	12
5.9 게임 화면 – 게임 결과 화면.....	13
6. 후기.....	13

## 1. 개요

추상화 과정은 소프트웨어 개발의 전반적인 부분에 걸쳐 흔하게 쓰이지만 가장 어려운 개념 중에 하나일 것이다. 아마 추상화라는 개념 자체가 상황에 따라 의미가 변할 수 있어 말이나 글로 설명하기는 모호한 개념이기 때문이라고 생각한다.

1학년 교육과정에서 우리는 절차지향언어인 C언어를 학습했었다. 절차지향언어는 이름에서 알 수 있듯 절차를 생각하며 코딩을 하는 방법으로 실로 1학년때 만들었던 C언어 프로젝트를 하나의 소스파일에 1200라인을 넘게 써서 프로그램을 구현하였던 것으로 기억한다.

이러한 방법은 틀리지는 않았다. 하지만 자바와 같은 객체지향언어로 추상화가 잘 되어있는 구조와 비교했을 때 전자의 코드는 코드의 재사용, 변수의 재사용 등으로 인해 가독성이 크게 떨어져 작성자가 보기엔 이해가 될 수 있으나 그 이외의 사람이 보기엔 이해가 잘 안되고 더 이상 수정, 추가를 할 수 없다고 생각하였다.

본 레포트에서는 이러한 추상화 과정을 염두에 두고 트럼프카드로 할 수 있는 게임인 블랙잭을 자바로 직접 처음부터 끝까지 구현하였다. 객체를 구상함에 있어 개념적인 부분을 통합, 분류하려 노력하였고 그렇게 구성된 객체 관계 구조로 확장성이 높은 코드를 구현할 수 있었으며 이를 이용하여 추후 더 많은 기능을 구현 해 볼 예정이다.

## 2. 블랙잭의 상세 규칙

### 2.1 카드 점수의 계산

블랙잭의 점수 계산에는 특별한 룰이 있다. 첫째는 Ace는 1 또는 11 양쪽 모두 될 수 있으며 J, Q, K는 10으로 계산되어 두 장으로 조합할 수 있는 최대 점수의 값이 21이 된다.

### 2.2 게임 방법

- 1) 베팅을 한 후 모든 플레이어와 딜러는 두 장의 카드를 받습니다.
- 2) 딜러는 자신의 카드 중 한 장을 엽니다.
- 3) 딜러는 카드의 합이 17이 될 때까지 반드시 추가 카드를 뽑아야 합니다.
- 4) 플레이어는 카드의 합이 21을 넘지 않는 범위 내에서 추가 카드를 받을 수도(Hit), 받지 않을 수도(Stay)있습니다.

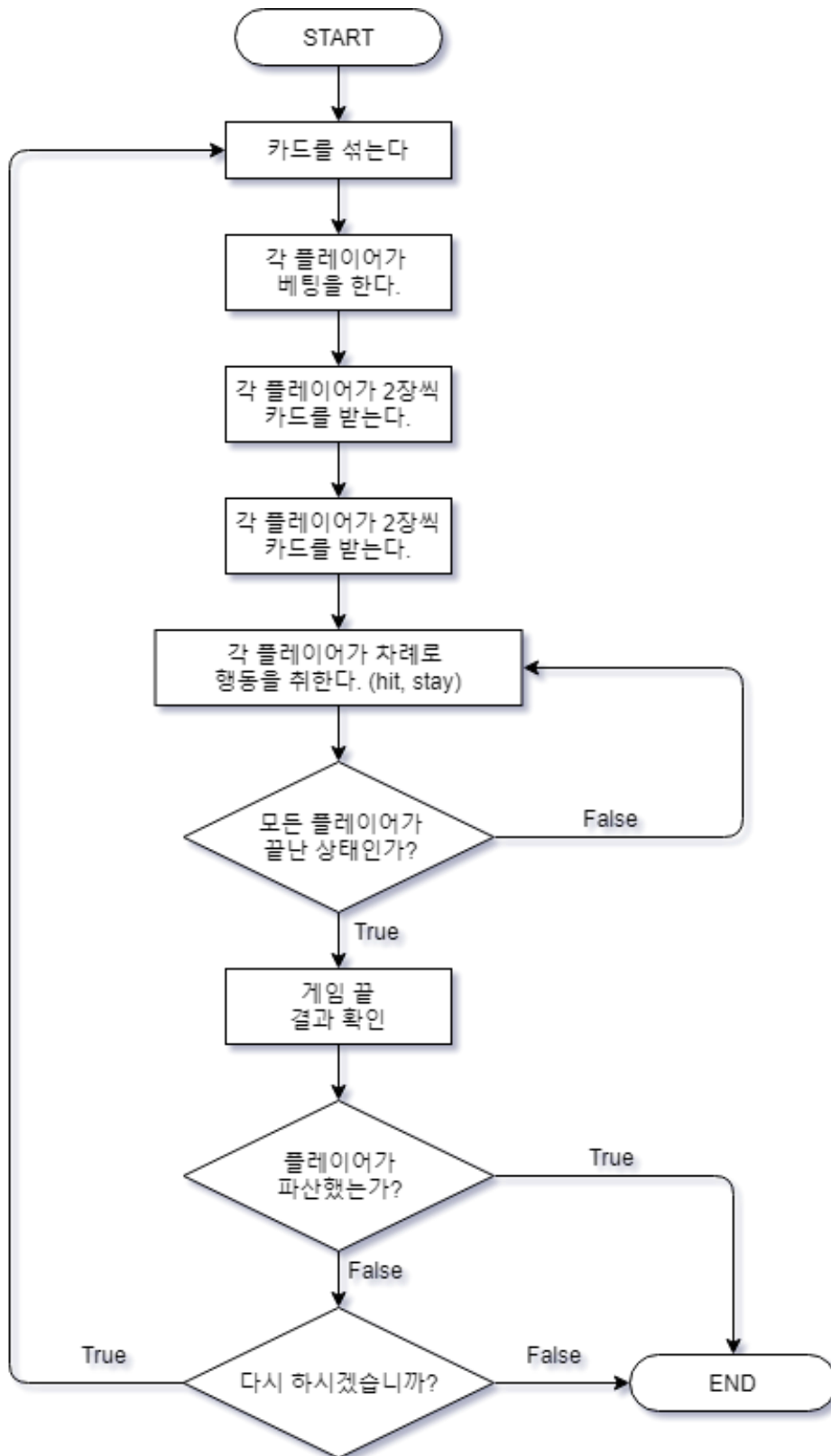
### 2.3 플레이어가 취할 수 있는 행동

- Hit : 플레이어가 처음 두 장의 카드 외에 딜러에게 추가카드를 요청하는 경우를 말합니다.
- Stay : 플레이어가 추가 카드를 원하지 않을 경우를 말하며, 딜러는 카드의 합이 17 이상이면 추가 카드를 받을 수 없습니다.
- Surrender : 플레이어가 처음 두 장의 카드로 딜러의 카드를 이길 수 없다고 판단한 경우 게임을 포기하는 것을 말합니다.

### 2.4 게임 결과

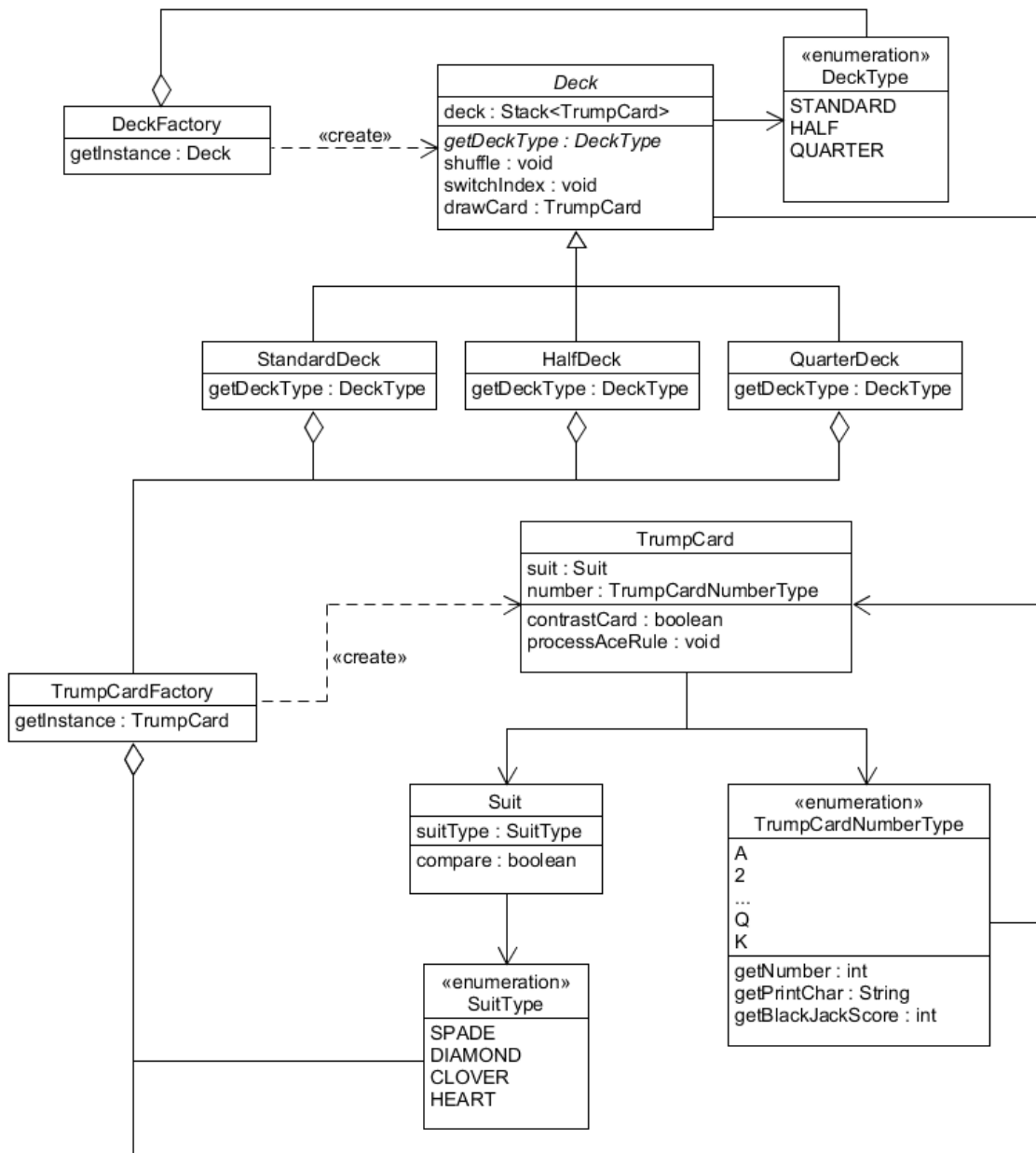
- Win : 딜러보다 높은 점수를 획득했을 시 베팅금액만큼 추가로 돈을 얻습니다.
- Lose : 딜러보다 적은 점수를 획득했을 시 베팅금액을 잃습니다.
- Bust : 플레이어의 점수가 21을 초과했을 시 결과와 상관없이 베팅금액을 잃습니다.
- Blackjack : 처음받은 2장의 점수가 21일 경우를 말하며 베팅 금액의 2배를 추가로 얻습니다.
- Surrender : 게임을 포기하고 베팅금액의 1/2를 돌려받습니다.

### 3. 블랙잭 게임 진행 순서도



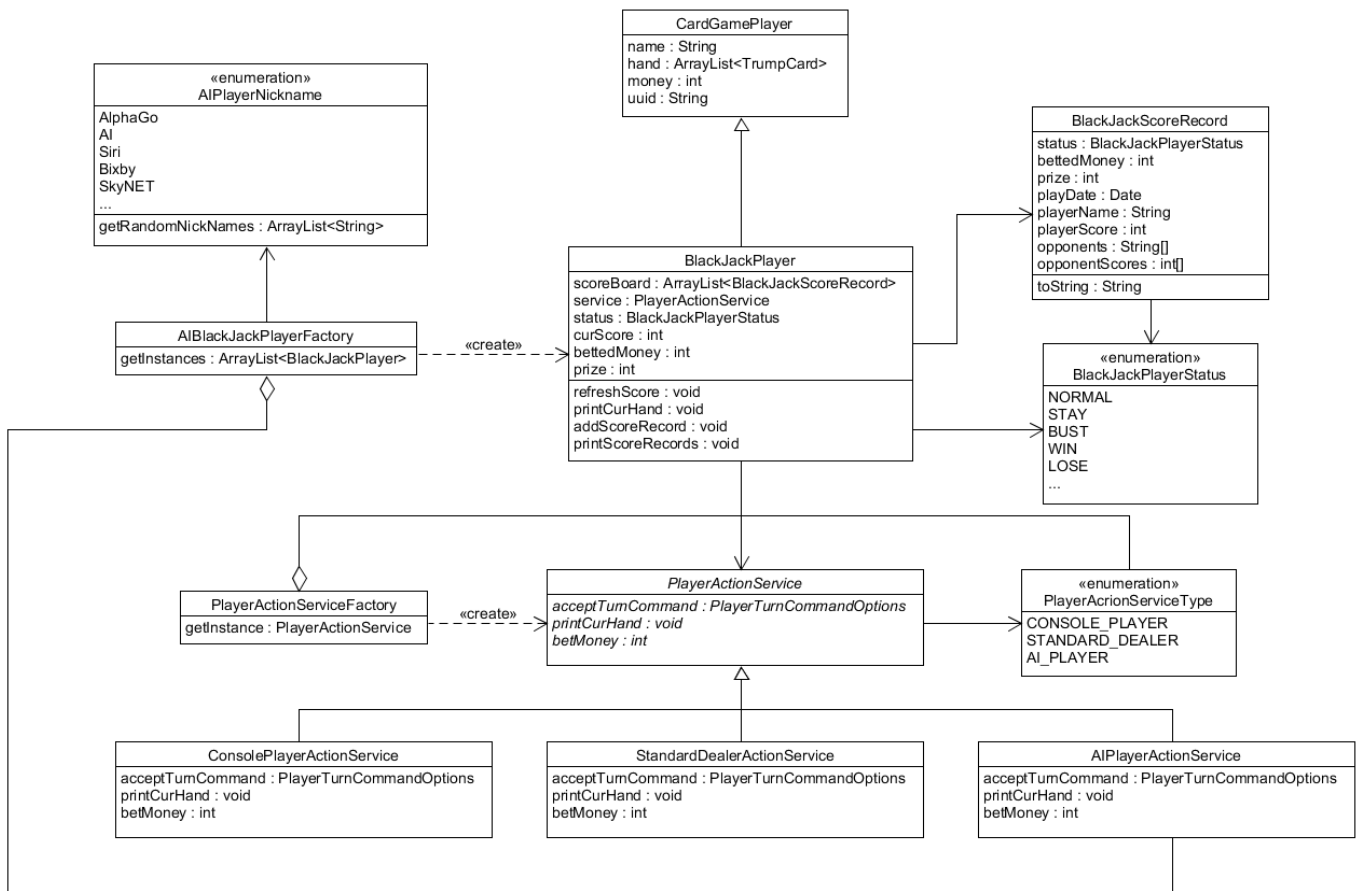
## 4. 클래스의 구현

### 4.1 카드와 카드덱의 구현



트럼프카드의 일반적인 점수와 블랙잭에서 사용되는 점수가 다르기에 블랙잭에서 사용하는 점수를 저장 해 두고 가져올 수 있게 구성하였다. 그리고 트럼프를 사용하여 덱을 만들도록 하였고 필요하다면 새로운 덱의 조합을 만들어 추가 할 수 있도록 덱을 인터페이스로 묶어주었다. 그리고 실제 카드 사용 시 `DeckFactory.getInstance` 메서드만을 사용하여 완성된 전체의 카드덱을 사용할 수 있도록 하였다.

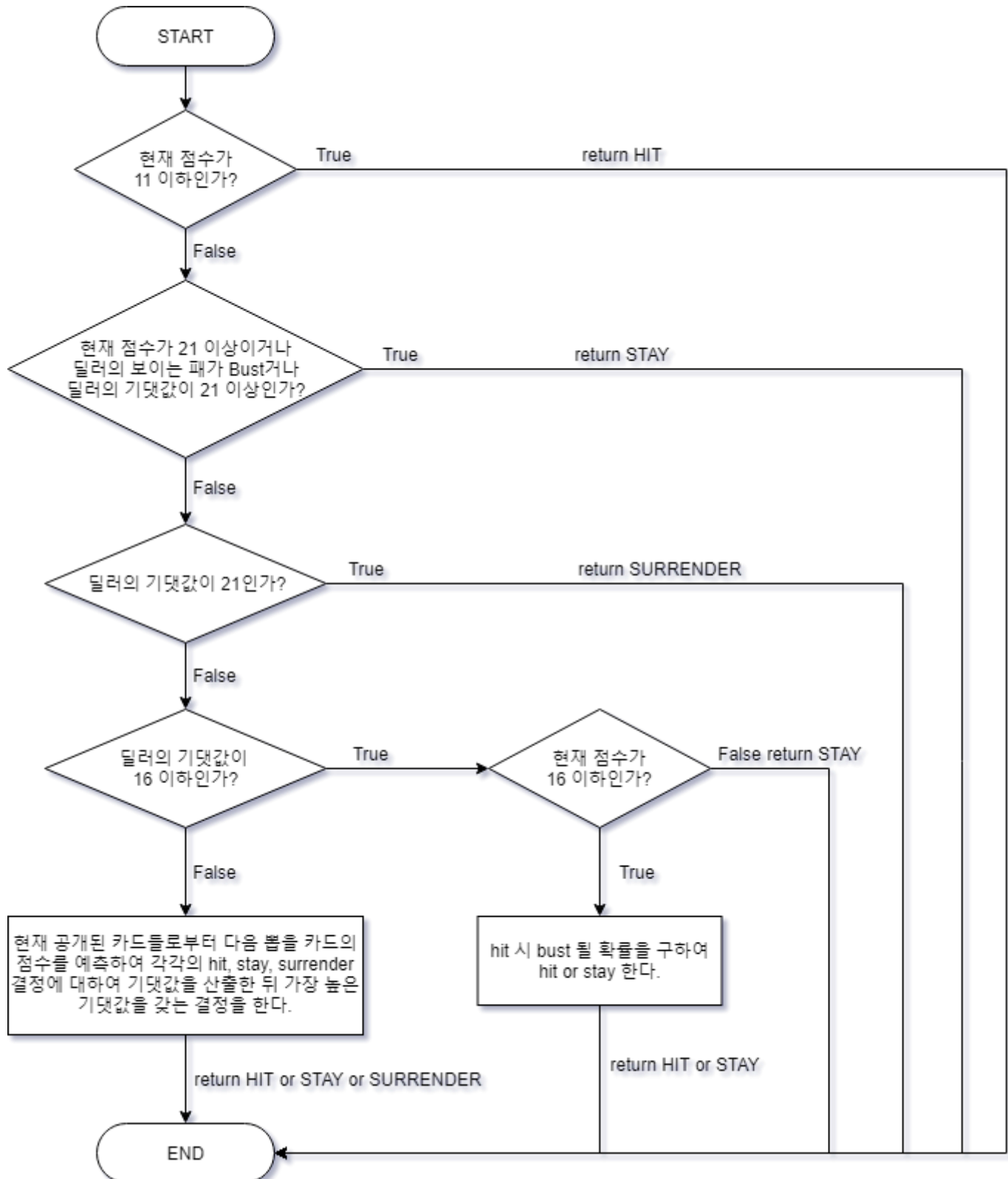
## 4.2 플레이어의 구현



플레이어가 가져야 하는 기본적인 부분을 구현하였다. 위의 그림에는 포함되어있지 않지만 BlackJackPlayer는 CardGamePlayer 클래스를 상속받아 name, hand, money, id의 필드가 생성되어있다. 플레이어의 현재 게임진행 상태를 enum값으로 저장하고 플레이어가 취하는 행동 패턴을 PlayerActionService로 구현해 여러 종류의 개성있는 플레이를 하는 플레이어들을 구현 할 수 있도록 구조를 짰다. 현재 이를 이용하여 실제 플레이어, 딜러, AI 플레이어의 행동패턴을 구현하였으며 AI 플레이어가 취하는 행동 알고리즘은 다음과 같다.

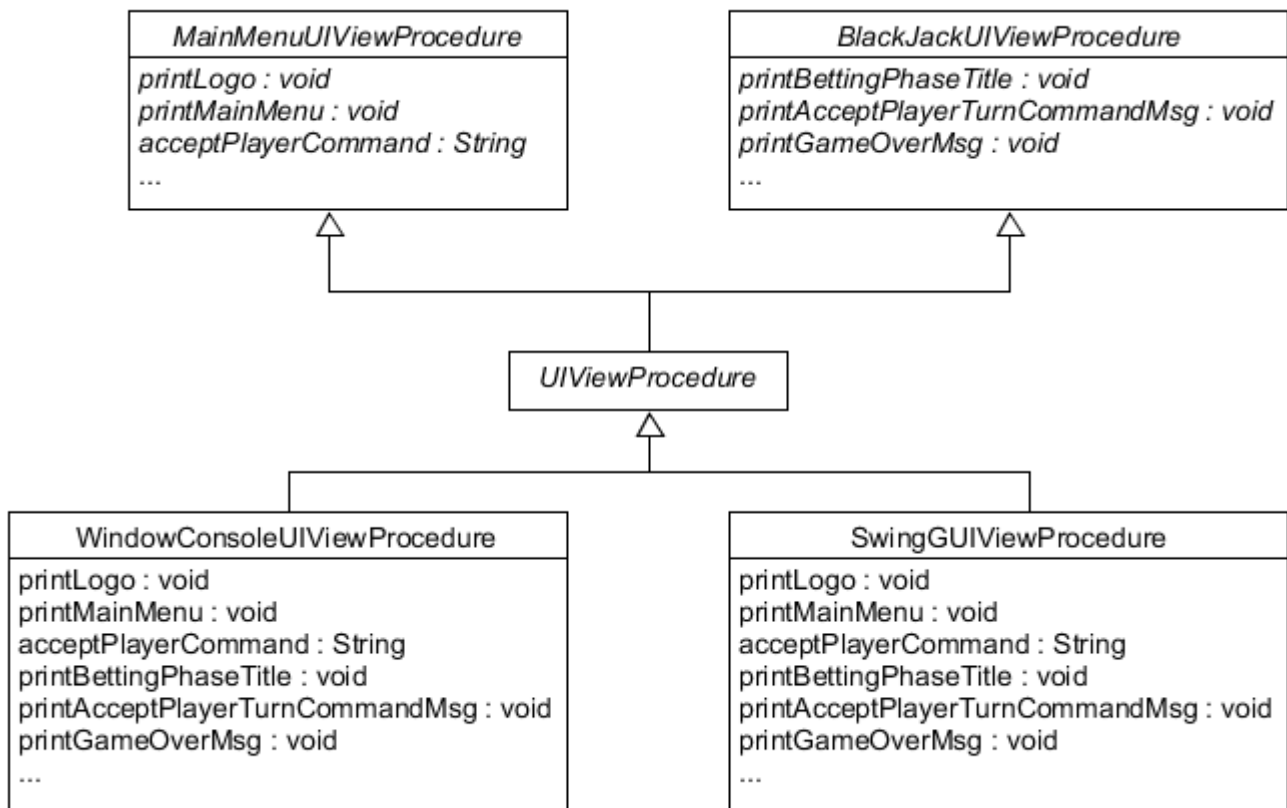
## 4.3 블랙잭 AI 플레이어 행동 패턴 구현

### 4.3.1 AI 플레이어 행동 패턴 알고리즘





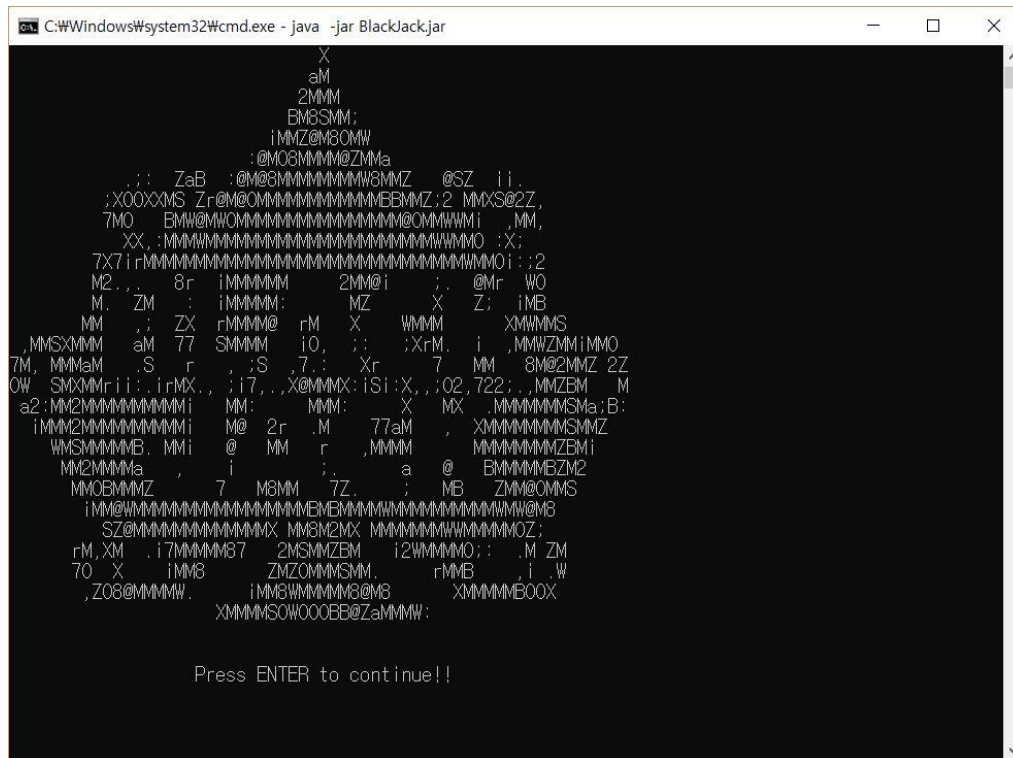
#### 4.4 UI를 출력하는 클래스의 구현



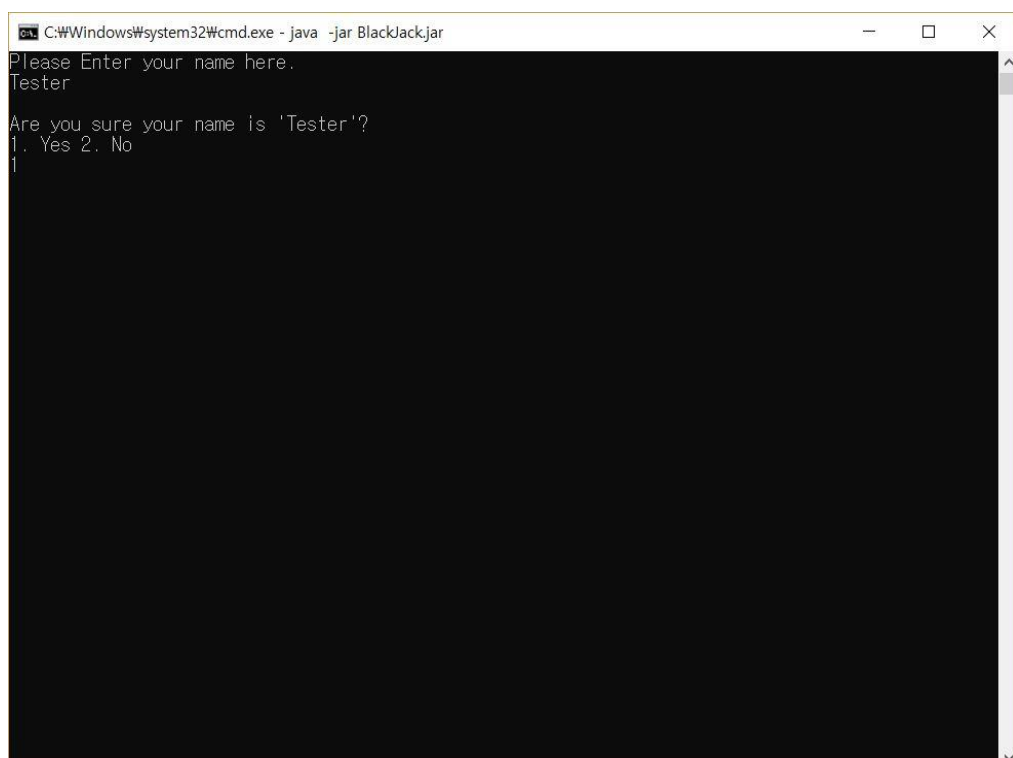
UI를 출력하는 부분을 인터페이스로 구현하여 여러 형태의 UI형태를 구현 할 수 있도록 하였다. 현재는 Windows 환경의 console창에서 출력하는 WindowConsoleUIViewProcedure만을 구현하였지만 추후 Java GUI 환경인 Swing으로 출력할 수 있게 SwingGUIUIViewProcedure도 구현해 보려고 한다.

## 5. 결과

## 5.1 게임 처음 실행 시 나타나는 스플래시 화면



## 5.2 플레이어 이름을 설정하는 화면



## 5.3 메인 메뉴

```
C:\Windows\system32\cmd.exe - java -jar BlackJack.jar
<==== Main Menu =====>

1. Start BlackJack
2. How to play?
3. Score board
4. Exit

Please Enter your command here.
```

## 5.4 게임 방법 설명화면

```
C:\Windows\system32\cmd.exe - java -jar BlackJack.jar
--카드 점수 계산--
Ace는 1 또는 11로 계산합니다.
King, Queen, Jack은 각각 10으로 계산합니다.
그 외의 카드는 카드에 표시된 숫자로 계산합니다.

--게임 방법--
베팅을 한 후 모든 플레이어와 딜러는 두 장의 카드를 받습니다.
딜러는 자신의 카드 중 한 장을 엽니다.
딜러는 카드의 합이 17이 될 때까지 반드시 추가 카드를 뽑아야 합니다.
플레이어는 카드의 합이 21을 넘지 않는 범위 내에서 추가 카드를 받을 수도(Hit),받지 않을 수도(Stay)있습니다.

- Blackjack
처음 두 장의 카드 합이 21일 경우를 말하며 베팅 금액의 2배의 Pay를 받습니다.

- Bust
카드 합이 21을 초과하면 베팅 금액을 잃게 됩니다.

- Draw
플레이어와 딜러의 각각의 카드 합이 같을 경우 서로 비기게 됩니다.

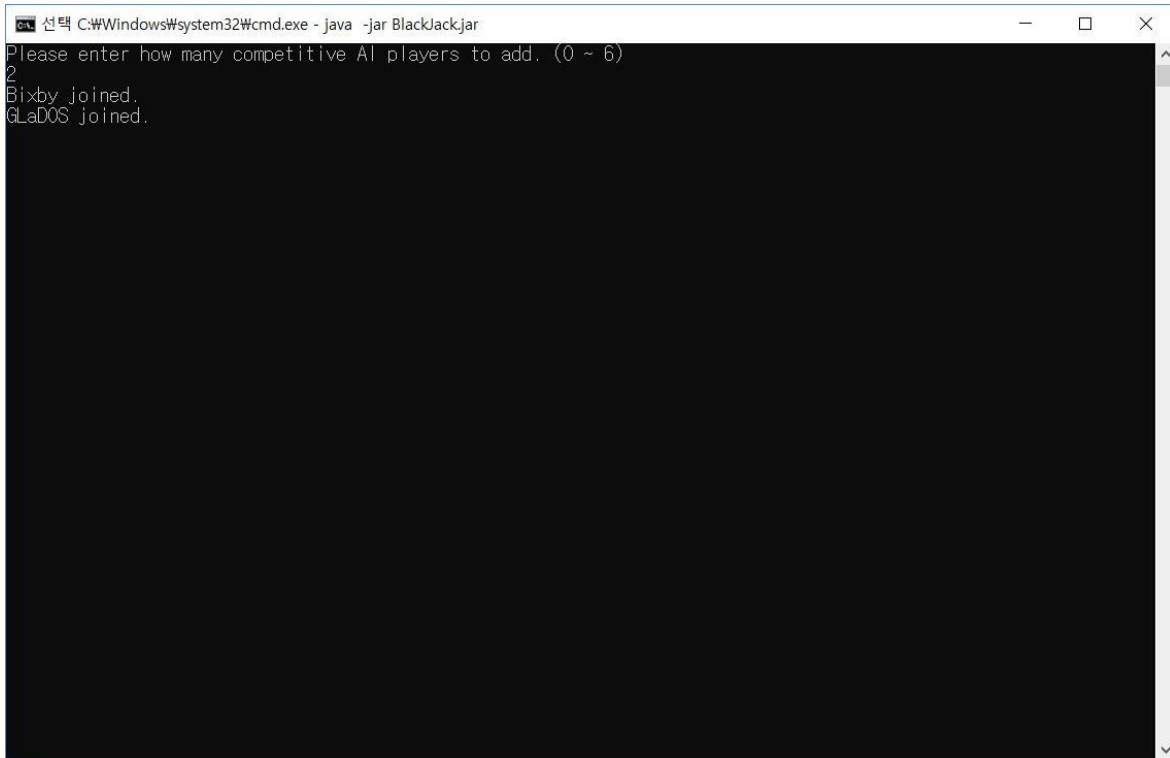
- Stay
플레이어가 추가 카드를 원하지 않을 경우를 말하며, 딜러는 카드의 합이 17 이상이면 추가 카드를 받을 수 없습니다.

- Hit
플레이어가 처음 두 장의 카드 외에 딜러에게 추가카드를 요청하는 경우를 말합니다.

- Surrender
플레이어가 처음 두 장의 카드로 딜러의 카드를 이길 수 없다고 판단한 경우 게임을 포기하는 것을 말합니다.
이 경우 최초 베팅액의 1/2을 잃게 됩니다.

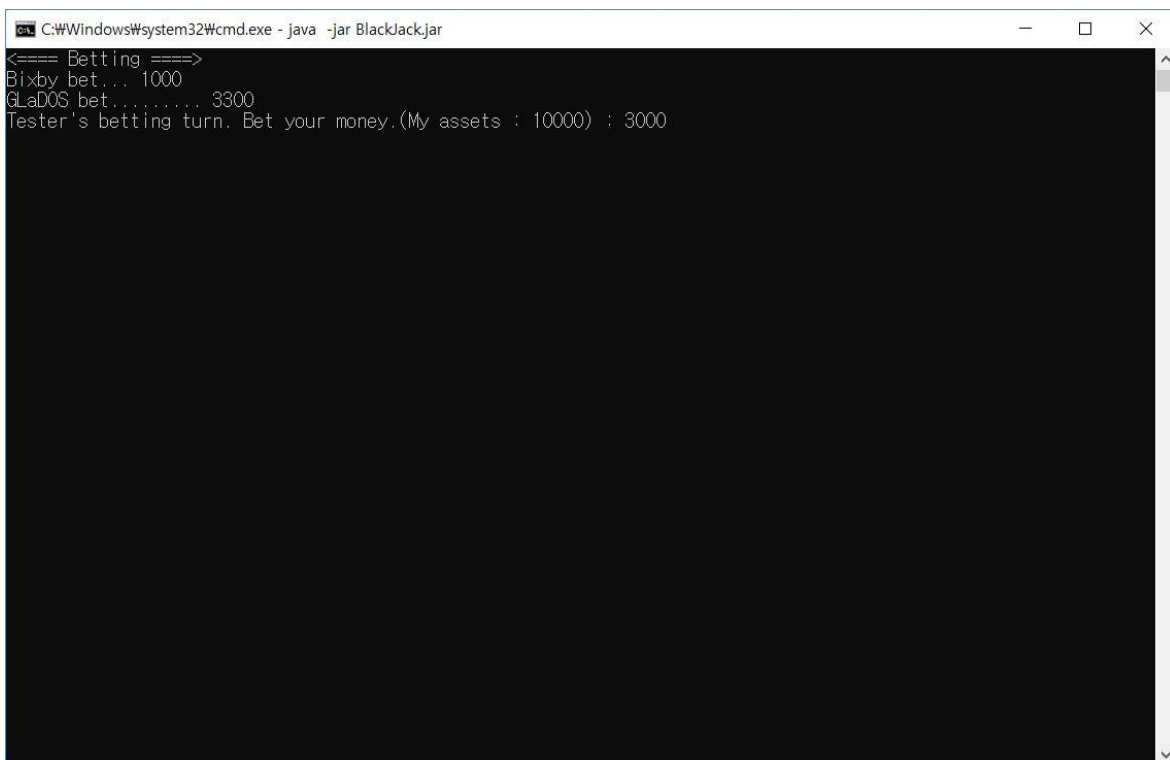
Press ENTER to continue!!
```

## 5.5 게임 시작 시 AI 플레이어 추가 화면



```
선택 C:\Windows\system32\cmd.exe - java -jar BlackJack.jar
Please enter how many competitive AI players to add. (0 ~ 6)
2
Bixby joined.
GLaDOS joined.
```

## 5.6 게임 시작 시 베팅화면



```
C:\Windows\system32\cmd.exe - java -jar BlackJack.jar
<==== Betting ====>
Bixby bet... 1000
GLaDOS bet..... 3300
Tester's betting turn. Bet your money.(My assets : 10000) : 3000
```

## 5.7 게임 화면 – AI 플레이어 턴

```
C:\Windows\system32\cmd.exe - java -jar BlackJack.jar
----- Dealer's Hand (? + 11/21) -----
[Redacted] 5 6

----- Bixby's Hand (? + 4/21) -----
Betted/Assets : 1000/10000
[Redacted] 4

----- GLaDOS's Hand (? + 10/21) -----
Betted/Assets : 3300/11000
[Redacted] J

----- Tester's Hand (20/21) -----
Betted/Assets : 3000/10000
Q K

GLaDOS's turn.....hit!!
```

## 5.8 게임 화면 – 플레이어 턴

```
C:\Windows\system32\cmd.exe - java -jar BlackJack.jar
----- Dealer's Hand (? + 11/21) -----
[Redacted] 5 6

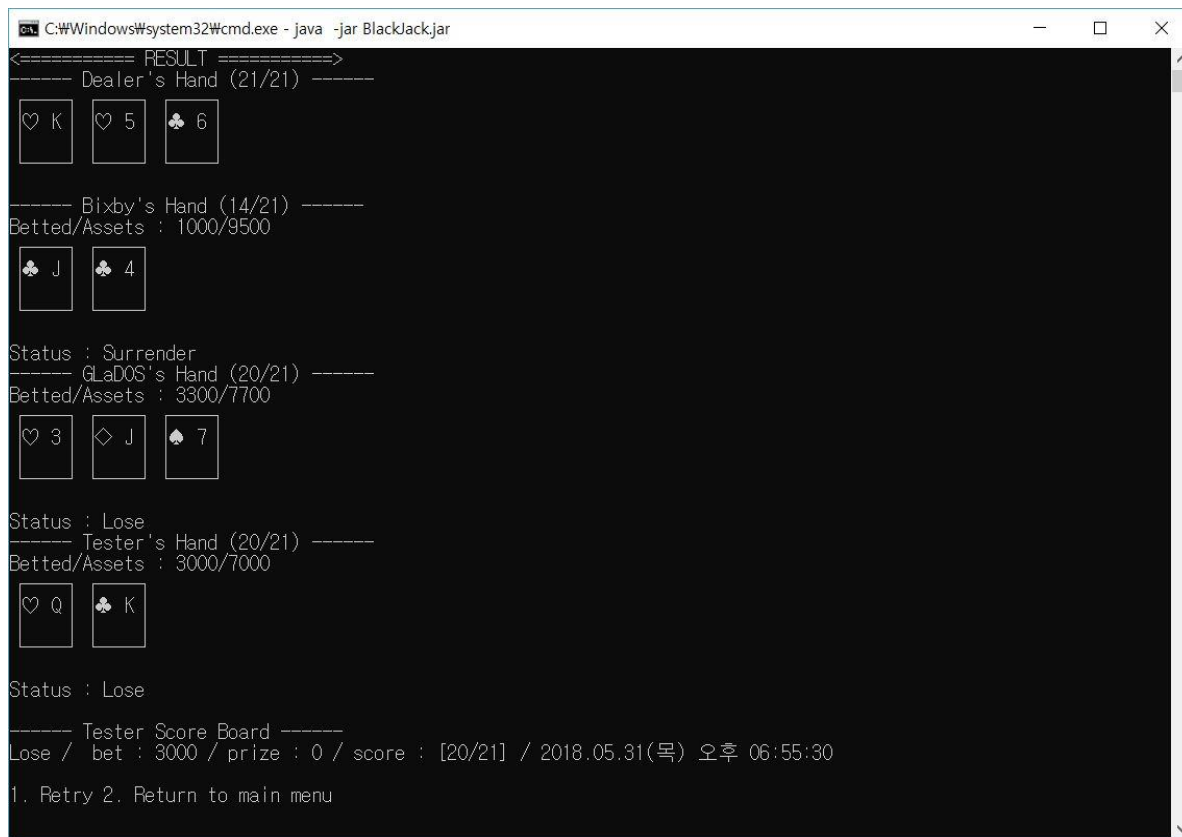
----- Bixby's Hand (? + 4/21) -----
Betted/Assets : 1000/10000
[Redacted] 4

----- GLaDOS's Hand (? + 17/21) -----
Betted/Assets : 3300/11000
[Redacted] J 7

----- Tester's Hand (20/21) -----
Betted/Assets : 3000/10000
Q K

Tester's turn. Make your decision..
1. Hit 2. Stay 3. Surrender
2
```

## 5.9 게임 화면 – 게임 결과 화면



```
<===== RESULT =====>
----- Dealer's Hand (21/21) -----
[♥ K] [♥ 5] [♣ 6]

----- Bixby's Hand (14/21) -----
Betted/Assets : 1000/9500
[♣ J] [♣ 4]

Status : Surrender
----- GLaDOS's Hand (20/21) -----
Betted/Assets : 3300/7700
[♥ 3] [♦ J] [♠ 7]

Status : Lose
----- Tester's Hand (20/21) -----
Betted/Assets : 3000/7000
[♥ Q] [♣ K]

Status : Lose

----- Tester Score Board -----
Lose / bet : 3000 / prize : 0 / score : [20/21] / 2018.05.31(목) 오후 06:55:30
1. Retry 2. Return to main menu
```

## 6. 후기

자바를 사용하여 블랙잭 게임을 구현하였다. 처음 시작 시 생각했던 것처럼 추상화를 염두에 두고 구조를 구상했고, 최대한 간결하게 짜려고 했으나 소스파일의 양이 생각보다 많아졌다. 하지만 각각의 객체가 추상화가 잘 되어 본 프로젝트에 추가하여 다른 게임방식의 블랙잭이나 심지어 아예 다른 카드게임도 구현 할 수 있을 정도의 수준이 되었다. 한가지의 문제를 해결하는 방법에 하나의 방법만 있는 것이 아니라는 것을 깨달을 수 있었고 유지보수, 기능확장을 염두에 두고 코드를 짜는 것이 중요하다는 것을 알게 되었다.