

1차 프로그래밍 과제

과목: 객체지향프로그래밍 및 실습

과제명: 1차 프로그래밍 과제

반: F089-1

이름: 정인아

학번: 202126546

제출일: 20230402

가) 소개

CardGame 프로그램은 종료 조건을 만족할 때까지 두 플레이어가 52장의 카드가 저장되어 있는 카드덱에서 각자 카드 두 장씩 뽑아 비교하고 진 사람이 이긴 사람에게 베팅 금액만큼 돈을 주는 게임 시스템이다. 게임이 시작되기 전 각 플레이어들에게 동일한 금액의 gameMoney가 배당되고 게임에서 지면 bettingMoney를 상대 플레이어에게 줘야 한다. 만약 플레이어의 gameMoney가 bettingMoney보다 작아지거나 카드 덱에 카드가 없으면 자동 종료된다. 한 베팅이 끝나면 게임을 지속할 것인지를 묻고 'y'라고 답하면 다음 베팅이 지속된다.

CardGame을 위해 필요한 클래스는 카드 정보를 담는 Card class, 플레이어로 기능하는 Player class, 카드덱으로써 기능하는 CardDeck class, 실제 게임이 진행되는 메인 클래스인 CardGame 클래스로 크게 나눌 수 있다. Player class에서는 상대 플레이어에게 bettingMoney를 주는 기능과 카드덱에서 카드 두장을 뽑는 기능이 있어야 한다. CardDeck class는 맨 위에 있는 카드를 뽑는 것 이기 때문에 Stack으로써 기능한다. 그러므로 top의 위치에서 카드를 빼고 넣는 함수가 필요하고 게임을 시작할 때 카드덱에 있는 카드를 섞기 위해 카드를 섞는 함수도 필요하다.

카드 게임의 주요 요구 사항은 게임 시작 전 카드덱 setting, 카드 비교, 게임 종료에 있다. 카드덱을 setting하기 위해서 카드52장을 생성하고 카드덱에 담는 initialize() 함수, 카드덱에 들어있는 카드 순서를 뒤섞는 shuffle() 함수를 구현해야 한다. 카드를 비교할 때 주의해야 할 사항은 하나의 카드끼리 비교했던 이전 실습과 달리 두 카드(hand)를 비교하기 때문에 고려해야 할 경우가 많아졌다. 따라서 아래와 같이 정리해보았다.

(one pair: 플레이어가 가진 두 카드의 rank가 같을 때를 말함)

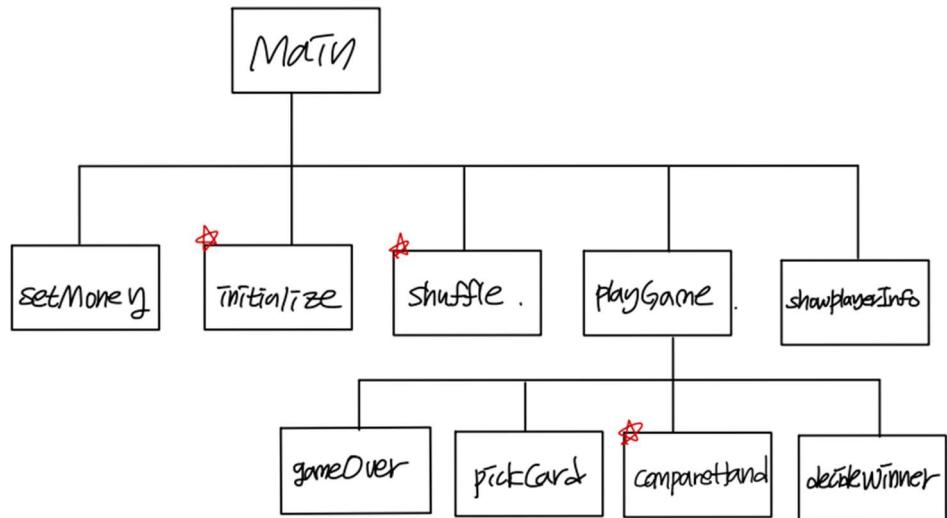
- ✓ 두 hand의 카드 비교는 어떻게 이루어지나?
 - 두 hand 중 한 hand 만 one pair일 경우 : one pair인 플레이어가 이긴다.
 - 둘 다 one pair 일 경우 : 두 hand의 rank 값을 비교한다.
 - 둘 다 one pair 가 아닌 경우
 - ① 두 플레이어가 지난 hand에서 rank값이 높은 카드끼리 비교한다.
 - ② 첫번째 비교에서 두 rank가 같았다면 두 플레이어가 지난 각 hand에서 작은 rank값을 가진 카드끼리 비교한다.
 - ③ 두 rank가 같다면 각 hand에서 높은 rank 를 가진 카드의 suit끼리 비교한다.

게임은 다음의 조건을 만족할 때 종료된다. 이는 if문을 이용하여 구현할 예정이다.

- ✓ 종료 조건:
 - ① 게임 머니가 베팅 금액보다 적을 때
 - ② 카드 덱에 카드가 없을 때

나) 분석 및 설계

- 모듈 차트



- 주요 알고리즘

Function : initialize

Algorithm: for i = 1 to 4 high

 for j = 1 to 13 high

 suit = i

 rank = j

 Convert the data type of the suit from int to String

 Put suit and rank into Card type object c

 Push c in CardDeck

 }

}

Function : shuffle

Algorithm:

// Creating a Random Number Array

for i = 0 to 52 high

 Pattern[i]=i

 for j = 0 to 52 high

 Select the two of random number ->randomNum1,randomNum2

 Swap pattern[randomNum1] and pattern[rankdomNum2]

// mix the cards according to this array(pattern)

```

for i = 0 to pattern.length high
    newS[i] = CardDeck.s[pattern[i]];
CardDeck.s = newS;

```

Function : compareHand

Input: Player p1, Player p2

Output: int (return 1 when p1 wins , return -1 when p2 wins, return 0 when tie)

Algorithm:

```

if isOnePair(p1) == true
    if isOnePair(p2) == false  return 1          // p1만 one pair일 경우
    else                      // 둘 다 one pair 일 경우
        result = compare Rank of p1 and p2
        if result>0 return 1
        else if result<0 return -1
        else return 0
    else
        if isOnePair(p2)==true  return -1         // p2만 one pair일 경우
        else                      //p1,p2 둘다 one pair일 경우
            c1,c2: a card with a high rank in the hand
            result = compare Rank of c1 and c2
            if result>0 return 1
            else if result<0 return -1
            else
                result = compare Rank of lowCardInHand(p1) and lowCardInHand(p2)
                if result>0 return 1
                else if result<0 return -1
                else
                    result= compare Suit of c1 and c2
                    if result>0 return 1
                    else if result<0 return -1
                    else return 0

```

다) 실행 결과

- 두 플레이어가 one pair 일 경우

```
Player 1 :(3 of Club, 3 of Diamond)
Player 2 :(2 of Club, 2 of Diamond)
player1 wins!
```

- 한 플레이어만 one pair 일 경우

```
Player 1 :(2 of Diamond, 2 of Heart)
Player 2 :(3 of Diamond, 10 of Heart)
player1 wins!
```

- 두 플레이어 다 one pair가 아닐 경우

- 각 hand에서 Rank가 높은 카드의 rank가 동일한 경우

```
Player 1 :(Jack of Diamond, Ace of Club)
Player 2 :(Ace of Heart, 8 of Club)
player1 wins!
```

- 각 hand에서 Rank가 낮은 카드의 rank도 동일한 경우

```
Player 1 :(10 of Diamond, 9 of Spade)
Player 2 :(9 of Club, 10 of Heart)
player1 wins!
```

- 각 hand에서 Rank가 높은 카드의 suit까지도 동일한 경우

→ 카드덤에 중복되는 카드가 없기 때문에 이 경우는 존재하지 않음

- 게임을 지속하지 않을 경우 (continue?에 'n'을 대답한 경우)

```
Games started...
Enter the game money:
100
Enter the betting money:
10
Player 1 :(2 of Diamond, King of Club)
Player 2 :(8 of Diamond, 3 of Club)
player1 wins!

continue?y
Player 1 :(7 of Diamond, 6 of Club)
Player 2 :(10 of Club, 10 of Heart)
player2 wins!

continue?y
Player 1 :(6 of Spade, 5 of Club)
Player 2 :(9 of Club, Ace of Club)
player2 wins!

continue?n
Game ended...
Total games: 3
Player 1 wins 1, and has $90
Player 2 wins 2, and has $110
```

- gameMoney가 100이고 bettingMoney가 10일 때 카드덱이 없어질때까지 실행한 결과

```
Games started...
Enter the game money:
100
Enter the betting money:
10
Player 1 :(King of Club, Queen of Club)
Player 2 :(4 of Club, 7 of Club)
player1 wins!

continue?y
Player 1 :(6 of Diamond, 8 of Club)
Player 2 :(5 of Club, Queen of Diamond)
player2 wins!

continue?y
Player 1 :(10 of Club, Jack of Spade)
Player 2 :(7 of Heart, 2 of Club)
player1 wins!

continue?y
Player 1 :(Jack of Heart, 9 of Heart)
Player 2 :(6 of Club, 3 of Heart)
player1 wins!

continue?y
Player 1 :(5 of Heart, 2 of Heart)
Player 2 :(Ace of Spade, 6 of Heart)
player2 wins!

continue?y
Player 1 :(Ace of Heart, 9 of Club)
Player 2 :(King of Spade, King of Heart)
player2 wins!

continue?y
Player 1 :(9 of Diamond, 10 of Spade)
Player 2 :(King of Diamond, 8 of Spade)
player2 wins!

continue?y
Player 1 :(2 of Diamond, 5 of Spade)
Player 2 :(6 of Spade, 4 of Heart)
player2 wins!

continue?y
Player 1 :(5 of Diamond, Queen of Spade)
Player 2 :(3 of Spade, 8 of Diamond)
player1 wins!

continue?y
Player 1 :(8 of Heart, 4 of Diamond)
Player 2 :(Ace of Diamond, 10 of Heart)
player2 wins!

continue?y
Player 1 :(3 of Diamond, Ace of Club)
Player 2 :(3 of Club, 9 of Spade)
player1 wins!
```

```

continue?y
Player 1 :(7 of Diamond, 7 of Spade)
Player 2 :(2 of Spade, 10 of Diamond)
player1 wins!

continue?y
Player 1 :(4 of Spade, Queen of Heart)
Player 2 :(Jack of Club, Jack of Diamond)
player2 wins!

continue?y
No more card!
Game ended...
Total games: 13
Player 1 wins 6, and has $90
Player 2 wins 7, and has $110

```

- 플레이어의 balance가 부족한 경우

```

Games started...
Enter the game money:
50
Enter the betting money:
25
Player 1 :(8 of Diamond, 6 of Diamond)
Player 2 :(5 of Spade, 9 of Spade)
player2 wins!

continue?y
Player 1 :(10 of Diamond, 3 of Spade)
Player 2 :(5 of Diamond, 3 of Diamond)
player1 wins!

continue?y
Player 1 :(3 of Heart, 4 of Club)
Player 2 :(8 of Heart, 9 of Diamond)
player2 wins!

continue?y
Player 1 :(10 of Spade, King of Heart)
Player 2 :(7 of Heart, 7 of Spade)
player2 wins!

continue?y
|Player1 doesn't have game money!
Game ended...
Total games: 4
Player 1 wins 1, and has $0
Player 2 wins 3, and has $100

```

라) 결론

이번 프로그래밍 과제로 CardGame 프로그램을 제작했다. 추가 요구 사항을 반영하기 위해 기존에 실습과제에서 구현했던 CardGame 프로그램에서 코드를 수정 및 추가하였다. 기존 실습에서 추가된 주요 요구 사항은 hand라는 개념과 Card를 담는 CardDeck이 생겼다는 것이다.

이전 실습과 달리 각 플레이어가 한 카드가 아닌 두 카드를 고르고 그 두 카드의 집합을 hand라고 한다. 두 hand (총 네 개의 카드) 를 hand 비교 규칙에 따라 비교해야 하므로 고려해야 하는 사항이 이전 실습에 비해 많아졌다. 처음에 플레이어가 가진 각 hand가 onepair인지 확인하고 둘 중 하나만 one pair이면 onepair인 플레이어가 이긴다. 둘 다 onepair이면 두 hand의 rank끼리 비교한다. 둘 다 one pair가 아니면 각 hand에서 rank가 높은 카드끼리 rank값을 비교하고 이 비교에서 '동등'이라는 결과가 나오면 각 hand에서 rank가 낮은 카드끼리 rank값을 비교한다. 이 비교에서 또 '동등'이라는 결과가 나오면 각 hand에서 rank가 높은 카드끼리 suit값을 비교한다. 이런 비교 과정을 구현하기 위해 if-else if-else 조건문을 다중 이용했다. 조건문을 다중으로 이용하다 보니 구현하면서 포함관계가 헷갈려 어려움을 겪었다. 이를 통해 if-else if-else 이용법과 개념을 확실히 알 수 있었다. 아쉬웠던 점은 hand를 비교하는 함수를 if-else if-else 조건문이 아닌 switch 문으로 했다면 한 눈에 알아보기 편해서 어려움이 덜 했을 것 같다.

게임 시작 전에 suit 4개와 rank 13개를 조합하여 만들 수 있는 모든 카드(개수: 52개)를 생성하여 CardDeck에 저장하고 무작위로 카드를 섞는 알고리즘을 짜야 한다. 이 알고리즘을 initialize(),shuffle()로 나누어 구현했다. 함수 initialize()는 suit와 rank를 조합하기 위해 이중 for문을 이용하고 suit와 rank순대로 Card []배열에 저장한다. 관건은 shuffle()함수이다. 정수형 배열에 1부터 52까지의 숫자를 크기 순으로 담고 배열 순서를 뒤섞어 랜덤한 패턴을 만든다. (배열 순서를 뒤섞을 때 math.random()함수가 이용된다.) 그 랜덤한 패턴을 앞서 suit와 rank순대로 저장된 Card []배열의 index값에 순차적으로 넣어 카드 순서가 뒤섞인 새로운 Card[]배열을 만든다. 이로써 shuffle 된 cardDeck을 얻을 수 있었다. Shuffle()함수를 제작함으로써 임시저장공간인 temp를 이용하여 두 변수의 값을 교환시키는 방법을 알게 되었다.