컴프2 3주차 : 블랙잭 구현

과제 목표 : 제네릭과 컬렉션을 활용하여 블랙잭 게임을 구현한다.

학번 : 201902722

이름 : 유형곤

큰 과제 : 블랙잭 구현하기

작은 과제 : Player 클래스 구현하기

```
public class Player {
    private ArrayList<Card> hand;

    public Player() {
        hand = new ArrayList<Card>();
    }

    public ArrayList<Card> getHand(){
        return hand;
    }

    public void hit(Card card) {
        hand.add(card);
    }
```

플레이어 클래스의 멤버변수는 ArrayList<Card> hand로, 플레이어가 소유한 카드를 의미합니다.

Public Player(): 생성자로, hand 변수를 초기화합니다.

Public ArrayList<Card> getHand() : hand를 반환합니다.

```java
public int getScore() {
    int sum = 0;
    Iterator<Card> iter = hand.iterator();
    while(iter.hasNext()) {
        String rank = iter.next().getRank();
        int score = 0;
        if(rank.equals("A")) {
            score = (sum > 10) ? 1 : 10;
        } else if(rank.equals("J") || rank.equals("Q") || rank.equals("K")) {
            score = 10;
        } else {
            score = Integer.parseInt(rank);
        }
        sum += score;
    }
    return sum;
}
```

Public int getScore() : 플레이어의 hand 안에 있는 카드들의 점수를 합하여 반환합니다.

숫자 카드에는 숫자만큼의 점수가, J, Q, K는 10점을, A는 카드의 합에 따라 1, 11의 점수를 부여합니다. 이 때 카드를 순회하기 위하여 Iterator를 사용하였습니다.

```java
public boolean isBust() {
    return (getScore() > 21);
}

public boolean isBlackJack() {
    return (getScore() == 21);
}
```

Public boolean isBust() : 현재 점수를 확인하여 21점보다 큰지 (BUst인지) 확인합니다.

Public boolean isBlackJack() : 현재 점수를 확인하여 21점인지 (BlackJack인지) 확인합니다.

D

작은 과제 : Card 클래스 구현하기

```java
public class Card {
    private String suit;
    private String rank;

    public Card(String suit, String rank) {
        this.suit = suit;
        this.rank = rank;
    }

    public String getSuit() {
        return suit;
    }

    public String getRank() {
        return rank;
    }

    @Override
    public String toString() {
        return String.format("[%s %s]", suit, rank);
    }

}
```

Card 클래스는 카드를 저장하기 위한 클래스로, 멤버변수로는 suit과 rank가 있습니다.

Public Card(String suit, String rank) : rank, suit을 초기화합니다.

Public String getSuit() : 카드의 suit을 반환합니다.

Public String getRank() : 카드의 rank를 반환합니다.

Public String toString() : Object 클래스의 toString 메서드를 오버라이드하여 카드를 문자열로 출력할 수 있도록 합니다.

작은 과제 : CardDeck 클래스 구현하기

```
public class CardDeck {
    private static final String suits[] = {"Spade", "Heart", "Diamond", "Clob"};
    private static final String ranks[] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "K
    private ArrayList<Card> deck;

    //카드 52장인 덱을 생성
    public CardDeck() {
        deck = new ArrayList<Card>();
        for(int i = 0; i < suits.length; i++) {
            for(int j = 0; j < ranks.length; j++) {
                Card card = new Card(suits[i], ranks[j]);
                deck.add(card);
            }
        }
    }
```

CardDeck 클래스는 말 그대로 덱 역할을 하는 클래스로, 앞서 만든 Card를 저장할 수 있도록 설계된 클래스입니다. 멤버변수에는 ArrayList<Card> deck (카드를 저장하는 기능)와 String suits[] (카드의 모든 suit을 저장), String ranks[] (카드의 모든 rank를 저장)이 있습니다.

Public CardDeck():

CardDeck의 생성자로, deck을 초기화하고 카드를 deck에 추가합니다.

```
//제일 앞에 있는 카드를 뽑는다.
public Card popCard() {
    return deck.remove(0);
}

//덱의 크기 반환
public int getSize() {
    return deck.size();
}

//덱을 섞는다.
public void shuffle() {
    Collections.shuffle(deck);
}
```

Public Card popCard() : 덱의 가장 위에있는 카드를 뽑습니다.

Public int getSize() : 덱에 남은 카드 수를 반환합니다.

Public void shuffle() 컬렉션 프레임워크를 통해서 deck을 섞습니다.