객체지향프로그래밍 – 4주차 실습 활동지 (2020년 9월 23일)

성명: _____ 학과: _____ 학번: _____ 반/그룹: _____

※ 본 실습활동지를 작성함에 있어 다른 학생의 문서로부터 일부 또는 전체를 복사하였습니까?
예( ) 아니오( ) (복사 하였다면 예에 체크하고 아니라면 아니오에 체크하시오)

제출방식: 별도의 답안지 문서에 각 문제에 대하여 작성한 답안이나 작성한 코드 및 그 실행 결과를 복사해 넣는다. 답안지 문서명은 "report_주차_학번" (예, report_4_201811111)로 만든다. 답안지는 pdf형식으로 저장하여 과제게시판(Bb)에 차주 월요일 23시59분까지 제출한다.

I. Objectives

1. Class를 정의하고 object를 생성하여 사용하는 기본 방법을 이해한다.
2. Java method를 정의하는 방법을 이해한다.
3. Primitive type과 reference type에 대해 이해하고 parameter passing에 활용할 수 있다.
4. Java class file구조에 대해 이해한다.

II. Exercises (15점)

1. 아래 코드 Card class를 이용하여 물음에 답하시오.

```
// Card.java
public class Card
{
    private int rank;
    private String suit;

    public Card(int r, String s)
    {
        rank = r;
        suit = s;
    }
    public int getRank()
    {
       return rank;
    }
    public String getSuit()
    {
       return suit;
    }
}
```

가) 다음 CardTest1.java를 실행해보시오. 결과는 무엇인가? **[실행화면 포함] (1점)**

```
// CardTest1.java
public class CardTest1
{
    public static void main(String[] args)
    {
        Card c1 = new Card(1, "Spade");
        Card c2 = new Card(2, "Heart");
        System.out.println("Suit : "+c1.getSuit()+", "+"Rank: "+c1.getRank());
        System.out.println("Suit : "+c2.getSuit()+", "+"Rank: "+c2.getRank());
    }
}
```

나) Card의 getRank() 및 getSuit()를 아래와 같이 수정한 후 CardTest1을 실행해보시오. 어떤 결과가 나오는가? (가)번과 결과를 비교해서 왜 그런지 설명하시오.**[실행화면 포함] (1점)**

```
public int getRank()
{
    return this.rank;
}
public String getSuit()
{
    return this.suit;
}
```

다) Card 객체의 카드 값을 아래와 같은 형식의 String으로 반환하는 toString() method를 Card class에 추가하시오. **[toStrnig() 메소드 및 실행화면 포함] (3점)**

Ace of Spade **//** 1 – Ace, 11 – Jack, 12 – Queen, 13 – King
2 of Heart

```
// Card.java
public class Card
{
    …
    public String toString()
    {
      // fill in the code here
    }
}
```

```
// CardTest2.java
public class CardTest2
{
    public static void main(String[] args)
    {
        Card c1 = new Card(1, "Spade");
        Card c2 = new Card(2, "Heart");
        System.out.println(c1.toString());
        System.out.println(c2.toString());
    }
}
```

라) (다)에서 작성한 Card class에 **compareTo()** method를 추가하고자 한다. compareTo() method는 하나의 Card object를 매개변수로 받아 자신과 비교하여 자신보다 크면 -1, 동일하면 0, 더 작으면 1을 반환한다. 아래 주어진 CardTest3.java를 이용하여 compareTo()가 제대로 작동하는지 확인하시오. 단, 카드의 순위는 2~3주차 실습과제와 동일하다.
(Hint: 2~3주차 실습과제에서 작성한 코드 부분을 활용) **[compareTo() method 및 실행화면 포함] (3점)**

```
public class Card
{
    …
    public int compareTo(Card other)
    {
      // fill in the code here
    }
}
```

힌트) 메소드 내에서 참조하는 instance variable은 그 메소드를 실행하는 객체의 instance variable임. 타 객체(예, 매개변수로 전달되는 객체)의 instance variable을 접근하기 위해서는 method를 호출한다. 예) rank > other.getRank() 에서 rank는 this.rank의 의미임.

```
// CardTest3.java
public class CardTest3
{
    public static void main(String[] args)
    {
        Card c1 = ...  // Card 객체를 생성할 것
        Card c2 = ...  // Card 객체를 생성할 것
        System.out.println(c1.toString());
        System.out.println(c2.toString());
```

```
            int result = c1.compareTo(c2);
            if(result > 0)
              System.out.println(c1.toString()+" wins "+c2.toString());
            else if(result < 0)
              System.out.println(c2.toString()+" wins "+c1.toString());
            else
              System.out.println(c1.toString()+" ties to "+c2.toString());
        }
    }
```

2. 1번 문제에서 정의한 Card class (Card.java)를 이용하여 Card object 52개를 가지는 CardDeck class를 정의하고자 한다. 다음 물음에 답하시오.

가) CardDeck class에서 Card 52장을 생성하여 초기화하는 constructor와 shuffle() method, card deck에서 순서대로 하나의 Card object를 반환하는 dealCard() method를 완성하시오. (Hint: 2-3주차 실습 과제에서 작성한 코드 부분을 응용) **[CardDeck.java 코드 포함] (5점)**

```
// CardDeck.java
public class CardDeck
{
    private static final int N = 52; // card의 수
    private Card[] deck;
    private int top=0;              // card deck의 현재 deal할 card 위치
    static final String[] cardSuits = {"spade", "diamond", "heart", "club"};

    public CardDeck()  // 카드 52장을 생성한다.
    {
        int k = 0;
        deck = new Card[N];
        for(int i=0; i<4; i++)
            for(int j=0; j<N; j++)
                deck[k++] = new Card(j, cardSuits[i]);
    }
    public void shuffle() // card deck을 무작위로 섞는다.
    {
       // fill in the code here
    }
    public Card dealCard()  // top위치에 있는 한 장의 카드를 반환한다.
    {
        // fill in the code here
    }
}
```

나) 문제1의 Card class와 2-가)의 CardDeck class를 이용하는 CardGame.java를 실행한 후, 그 결과를 설명하시오. **[실행화면 및 결과 설명] (2점)**

```
// CardGame.java
public class CardGame
{
    public static void main(String[] args)
    {
        CardDeck deck = new CardDeck();
        Card player1;
        Card player2;

        deck.shuffle();
        player1 = deck.dealCard();
        player2 = deck.dealCard();

        int result = player1.compareTo(player2);
        if(result > 0)
            System.out.println(player1.toString()+" wins "+player2.toString());
        else if(result < 0)
            System.out.println(player2.toString()+" wins "+player1.toString());
        else
            System.out.println(player1+" ties to "+player2);
    }
}
```