

객체지향프로그래밍 -3주차 실습 활동지 (2019년 9월 17일)

성명: 안창희 학과: 소프트웨어 학번: 201723272 실습호실: 318

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

1) 아래 그림과 같이 52장의 카드 값과 종류를 저장하기 위한 2개의 배열을 초기화 하는
함수initialize()를 적절하게 채우시오.

rank array

1	2	...	13	1	2	...	13	1	2	...	13	1	2	...	13
0	1		12	13	14		25	26	27		38	39	40		51

suit array

spade	spade	...	spade	diamond	diamond	...	diamond	heart	heart	...	heart	club	club	...	club
0	1		12	13	14		25	26	27		38	39	40		51

```
47 public static void initialize()
48 {
49
50
51     for (int i = 0; i < rank.length; i++)
52     {
53         rank[i] = i+1; //배열 rank의 원소에 각각 1부터 52에 해당하는 숫자 순서대로 임시지정
54         suit[i] = "spade"; //suit의 모든 원소를 spade 값으로 임시지정
55     }
56
57     for(int i =13 ; i <26 ; i++)
58     {
59         rank[i] = rank[i]-13; //rank[13]~rank[25]에 해당하는 숫자를 1~13으로 수정
60         suit[i] = "diamond"; //suit[13]~rank[25]에 해당하는 문자열을 diamond로 수정
61     }
62
63     for(int i =26 ; i <39 ; i++)
64     {
65         rank[i] = rank[i]-26; //rank[26]~rank[38]에 해당하는 숫자를 1~13으로 수정
66         suit[i] = "heart"; //suit[26]~rank[38]에 해당하는 문자열 heart로 수정
67     }
68
69     for(int i =39 ; i <52 ; i++)
70     {
71         rank[i] = rank[i]-39; //rank[39]~rank[51]에 해당하는 숫자를 1~13으로 수정
72         suit[i] = "club"; //suit[39]~suit[51]에 해당하는 문자열을 club로 수정
73     }
74
75 }
```

출력부 1

terminated: cardsame.java:app

카드 전체 목록을 출력합니다

종류 : spade 숫자 :1
종류 : spade 숫자 :2
종류 : spade 숫자 :3
종류 : spade 숫자 :4
종류 : spade 숫자 :5
종류 : spade 숫자 :6
종류 : spade 숫자 :7
종류 : spade 숫자 :8
종류 : spade 숫자 :9
종류 : spade 숫자 :10
종류 : spade 숫자 :11
종류 : spade 숫자 :12
종류 : spade 숫자 :13
종류 : diamond 숫자 :1
종류 : diamond 숫자 :2
종류 : diamond 숫자 :3
종류 : diamond 숫자 :4
종류 : diamond 숫자 :5
종류 : diamond 숫자 :6
종류 : diamond 숫자 :7
종류 : diamond 숫자 :8
종류 : diamond 숫자 :9
종류 : diamond 숫자 :10
종류 : diamond 숫자 :11
종류 : diamond 숫자 :12
종류 : diamond 숫자 :13
종류 : heart 숫자 :1
종류 : heart 숫자 :2

출력부2

terminated: cardsame.java:7

종류 : diamond 숫자 :12
종류 : diamond 숫자 :13
종류 : heart 숫자 :1
종류 : heart 숫자 :2
종류 : heart 숫자 :3
종류 : heart 숫자 :4
종류 : heart 숫자 :5
종류 : heart 숫자 :6
종류 : heart 숫자 :7
종류 : heart 숫자 :8
종류 : heart 숫자 :9
종류 : heart 숫자 :10
종류 : heart 숫자 :11
종류 : heart 숫자 :12
종류 : heart 숫자 :13
종류 : club 숫자 :1
종류 : club 숫자 :2
종류 : club 숫자 :3
종류 : club 숫자 :4
종류 : club 숫자 :5
종류 : club 숫자 :6
종류 : club 숫자 :7
종류 : club 숫자 :8
종류 : club 숫자 :9
종류 : club 숫자 :10
종류 : club 숫자 :11
종류 : club 숫자 :12
종류 : club 숫자 :13

2) 초기화한 52장의 카드를 뒤죽박죽 섞는 shuffle() 함수를 작성하시오. 힌트) 각 카드의 값과 종류가 동시에 교환되어야 함. Math.random() 함수를 사용할 것.

```

78
79
80 public static void shuffle()
81 {
82     int TempRank; //임시rank값
83     String TempSuit; //임시suit값
84
85     for(int i=0 ; i<rank.length ; i++) //0번째 칸부터 51번째 칸까지의 모든 원소에게 한번씩 실행
86     {
87         int RandomNum = (int) (Math.random() * 51); //임의의 51 이하의 정수를 RandomNum 에 입력
88
89
90         //i번째 칸 원소의 rank와 suit를 임시저장
91         TempRank = rank[i];
92         TempSuit = suit[i];
93
94         //i번째 칸 원소의 rank와 suit값을 RandomNum번째 값으로 대체
95         rank[i] = rank[RandomNum];
96         suit[i] = suit[RandomNum];
97
98
99         //RandomNum번째 원소를 아까 임시저장한 값으로 대체
100        rank[RandomNum] = TempRank;
101        suit[RandomNum] = TempSuit;
102
103    }
104    //결과적으로 1번째부터 52번째 원소를 차례대로 임의의 다른 원소와 Swap하는 함수이다
105 }
106
107

```

서품완료된 카드 전체 목록을 출력합니다

```

종류 : spade 숫자 :4
종류 : club 숫자 :8
종류 : diamond 숫자 :12
종류 : heart 숫자 :12
종류 : heart 숫자 :5
종류 : spade 숫자 :9
종류 : diamond 숫자 :13
종류 : diamond 숫자 :11
종류 : heart 숫자 :8
종류 : club 숫자 :13
종류 : spade 숫자 :6
종류 : spade 숫자 :2
종류 : spade 숫자 :8
종류 : heart 숫자 :11
종류 : club 숫자 :9
종류 : diamond 숫자 :9
종류 : diamond 숫자 :1
종류 : heart 숫자 :9
종류 : spade 숫자 :11
종류 : spade 숫자 :10
종류 : diamond 숫자 :8
종류 : diamond 숫자 :6
종류 : heart 숫자 :10
종류 : club 숫자 :10
종류 : diamond 숫자 :2
종류 : club 숫자 :3
종류 : spade 숫자 :5
종류 : diamond 숫자 :5
종류 : club 숫자 :5

```

```

종류 : club 숫자 :5
종류 : heart 숫자 :6
종류 : heart 숫자 :13
종류 : diamond 숫자 :3
종류 : heart 숫자 :1
종류 : spade 숫자 :3
종류 : heart 숫자 :2
종류 : diamond 숫자 :4
종류 : club 숫자 :1
종류 : club 숫자 :11
종류 : club 숫자 :6
종류 : club 숫자 :4
종류 : heart 숫자 :3
종류 : spade 숫자 :7
종류 : heart 숫자 :7
종류 : spade 숫자 :13
종류 : diamond 숫자 :7
종류 : spade 숫자 :12
종류 : diamond 숫자 :10
종류 : heart 숫자 :4
종류 : club 숫자 :2
종류 : spade 숫자 :1
종류 : club 숫자 :12
종류 : club 숫자 :7

```

3) 카드 52장을 저장된 순서대로 출력하는 함수 printCards() 를 작성하시오.

```
109
110 public static void printCards()
111 {
112     //배열[0]~배열[51]까지 해당하는 rank와 suit를 찍지어서 출력하는 함수이다.
113     for(int count=0 ; count < rank.length ; count++ )
114         System.out.println("종류 : " + suit[count] + " 숫자 : " + rank[count] );
115 }
```

4) 두 개의 카드의 값과 종류를 매개변수로 받아 앞 카드가 크면 0을, 뒤 카드가 크면 1을 반환하는 함수 compareCards()를 작성하시오.

compareCard 함수는 입력받은 두 개의 카드의 종류와 숫자를 비교하여 result값을 0과 1로 반환한다.

if~else if 문을 차례로 사용하여 a와 b의 숫자가 $a > b$ $a < b$ $a == b$ 의 순서대로 코드를 입력했고

$a > b$ $a < b$ 의 경우에는 바로 result값을 반환하면 되나 $a == b$ 의 경우에는 카드의 종류를 비교하는 것으로 구현했다.

a의 종류를 기준으로 하여 차례로 비교하는 것을 목표로 하여 작성했으며, 본인이 구상한 compareCards의 구현 규칙은 다음과 같다.

1. aSuit가 spade인 경우에는 바로 a가 승리
2. aSuit가 diamond인 경우에는, b가 spade이면 b의 승리 / b가 spade가 아니면 a가 승리
3. aSuit가 heart인 경우에는, b가 spade 혹은 diamond면 b의 승리 / 아니면 a가 승리
4. aSuit가 club인 경우에는 바로 b가 승리

의 알고리즘을 코드로 옮겼다.

compareCards() 구현코드

```
117 public static int compareCards(int aRank, String aSuit, int bRank, String bSuit)
118 {
119     int result=0; //결과 (앞카드가 크면 0 뒤카드가 크면 1로 약속함), 일단 0으로 초기화
120
121     //두 카드의 숫자를 비교
122     if(aRank > bRank) //a의 숫자가 클 경우 => a승리
123     {
124         result = 0;
125     }
126
127     else if(aRank < bRank) //b의 숫자가 클 경우 => b승리
128     {
129         result = 1;
130     }
131
132     else if(aRank == bRank) //a와 b의 숫자가 같은경우
133     {
134         if(aSuit == "spade") //a의 종류가 스페이드면 => a승리
135             result = 0;
136
137         if(aSuit == "diamond") //a의 종류가 다이아몬드일경우
138         {
139             if(bSuit == "spade") //b가 스페이드면 => b승리
140                 result = 1;
141
142             else //중복된 다이아몬드가 나올 경우는 없으므로 b가 스페이드가 아니면 => a승리
143                 result = 0;
144         }
145
146         if(aSuit == "heart") //a의 종류가 하트일경우
147         {
148             if(bSuit == "spade" || bSuit == "diamond") //b가 스페이드or다이아몬드면 => b승리
149                 result = 1;
150             else //b가 클럽인경우=> a승리
151                 result = 0;
152         }
153
154         else if (aSuit == "club") //a의 종류가 클럽일경우
155             result = 1; // => b의 승리
156     }
157     //두 카드의 종류를 비교
158
159     return result; //결과값을 출력
160 }
161
```


5) main()함수의 빈곳 A와 B를 적절하게 채우시오.

```
5
6 // A 비교할 카드 두개를 설정하고 assign
7 System.out.println("\n 셔플이 완료된 상태에서 맨 앞카드와 맨 뒷카드를 비교합니다");
8 aRank = rank[0];
9 aSuit = suit[0];
10 bRank = rank[51];
11 bSuit = suit[51];
12
13 System.out.println("\n" + aSuit + " " + aRank + " VS " + bSuit + " " + bRank);
14
15 result = compareCards(aRank, aSuit, bRank, bSuit);
16 // B 비교결과를 출력하는 부분
17 if(result == 0)
18     System.out.println("\n[0]번째 카드 " + aSuit + " " + aRank+ "의 승리입니다!");
19
20 if(result == 1)
21     System.out.println("\n[51]번째 카드 " + bSuit + " " + bRank+ "의 승리입니다!");
22
23
```

aSuit와 bSuit의 할당에서는 이미 cardShuffle에서 모든 카드목록의 카드가 랜덤으로 섞였으므로 입력값을 Scanner로 구현하는 것이 큰 의미가 없다고 판단하여 하드코딩으로 카드덱의 맨 앞과 맨 뒤 카드를 비교하도록 지정하였다.

비교결과 출력에서는 result값의 출력 이전에 맨 앞 카드와 맨 뒤 카드의 정보를 VS 형식으로 콘솔에 출력해주어 비교되는 두 카드의 정보를 사전에 파악할 수 있도록 코딩했으며 result 값에 따라 어느 카드의 승리인지를 출력하고 승리한 카드의 정보를 다시금 출력하도록 작성했다.

난이도가 높지는 않지만 C언어만 사용하던 본인에게 처음으로 java 언어를 사용하여 알고리즘을 설계하고 코드로 구현하며 설명하는 과제였고,

배열의 전체원소 할당을 for문으로 구현하는 부분에서 에러를 해결하기위해 교재 코드를 참고하는 중,

C와는 다르게 배열이름.length의 형식으로 구현하는 것을 보고 참고하여 코드를 수정해가며, 이론시간에 배웠던 객체와 클래스개념과, 자바 언어의 특성인 robust 등의 추상적인 개념을 조금은 이해할 수 있었다.

추가내용) 보고서 작성당시 compareCard() 부분에서 Ace가 King을 이기는 구현과 B 부분에서 1,11,12,13 의 숫자를 Ace, Jack, Queen, King 으로 출력하는 구현을 코드에서는 추가로 구현시켜서 제출하였다.