

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Лабораторна робота №9

з курсу:

“ОБ’ЄКТНО ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ”

Виконав:

ст. гр.

КН-110

Халус

Богдан

Прийняв:

Гасько Р.Т.

Львів – 2018 р.

Лабораторна робота № 9

Завдання: Розробка власних контейнерів.

Виконання роботи

GitHub link - https://github.com/bhalus/OOP_Java/tree/master/lab_9

```
Deck.java
1 import java.util.*;
2 import java.util.concurrent.ThreadLocalRandom;
3
4 public class Deck {
5
6     private int size;
7     private Card[] cards;
8     private int[] cardsVals;
9
10    public Deck () {
11        this.cards = new Card[36];
12        this.cardsVals = new int[36];
13        this.size = 36;
14
15        int j = 0;
16        int k = 0;
17
18        for (int i = 0; i < 36; i++) {
19            if (k == 0) {
20                k = 0;
21                j++;
22            }
23            this.cards[i] = new Card( Rank.values[k], Suit.values[j]);
24            this.cardsVals[i] = i;
25            k++;
26        }
27    }
28
29    public void shuffle () {
30
31        Random rnd = ThreadLocalRandom.current();
32        for (int i = 0; i < this.size; i++) {
33            int index = rnd.nextInt(i+1);
34            Card a = this.cards[index];
35            int aa = this.cardsVals[index];
36            this.cards[index] = this.cards[i];
37            this.cardsVals[index] = this.cardsVals[i];
38            this.cards[i] = a;
39            this.cardsVals[i] = aa;
40        }
41    }
42
43    public void order () {
44
45        int temp = 0;
46        Card tempCard;
47        for (int i = 0; i < this.size; i++){
48            for (int j = i; j < (this.size-1); j++){
49                if (this.cardsVals[j-1] > this.cardsVals[j]){
50                    //swap elements
51                    temp = this.cardsVals[j-1];
52                    tempCard = this.cards[j-1];
53                    this.cardsVals[j-1] = this.cardsVals[j];
54                    this.cards[j-1] = this.cards[j];
55                    this.cardsVals[j] = temp;
56                    this.cards[j] = tempCard;
57                }
58            }
59        }
60    }
61
62
63
64    public boolean hasNext () {
65        if (this.size > 0) {
66            return true;
67        } else {
68            return false;
69        }
70    }
71
72    public Card drawOne () {
73
74        this.size--;
75        return this.cards[this.size];
76    }
77
78
79    public void printallCards () {
80        for (int i = 0; i < this.size; i++) {
81            this.cards[i].printCard();
82        }
83    }
84
85    public void printSize () {
86        System.out.println(size);
87    }
88
89 }
```

```
Deck.java
47    Card tempCard;
48    for (int i = 0; i < this.size; i++){
49        for (int j = i; j < (this.size-1); j++){
50            if (this.cardsVals[j-1] > this.cardsVals[j]){
51                //swap elements
52                temp = this.cardsVals[j-1];
53                tempCard = this.cards[j-1];
54                this.cardsVals[j-1] = this.cardsVals[j];
55                this.cards[j-1] = this.cards[j];
56                this.cardsVals[j] = temp;
57                this.cards[j] = tempCard;
58            }
59        }
60    }
61
62
63
64    public boolean hasNext () {
65        if (this.size > 0) {
66            return true;
67        } else {
68            return false;
69        }
70    }
71
72    public Card drawOne () {
73
74        this.size--;
75        return this.cards[this.size];
76    }
77
78
79    public void printallCards () {
80        for (int i = 0; i < this.size; i++) {
81            this.cards[i].printCard();
82        }
83    }
84
85    public void printSize () {
86        System.out.println(size);
87    }
88
89 }
```