

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №3**  
з дисципліни  
«Об’єктно-орієнтоване програмування»

**Виконав:**

студент групи КН-108

Кіт Володимир Дмитрович

**Викладач:**

Гасько Р.Т.

Львів – 2018 р.

## Написання програм зв'язний список , колода карт, числа Фібоначчі

Зв'язний список:

```
package com.tasks3.linkedlist;
```

```
public class LinkedList {
    private Node tail;
    private Node head;
    private int size = 0;
    public static void main(String[] args) {

        LinkedList list = new LinkedList();
        list.add(1);
        System.out.println( list.get(0));
        System.out.println(list.size());
    }

    public LinkedList() {

    }

    public void add(Integer data) {
        Node newNode = new Node();
        newNode.setData(data);

        if (size == 0)
            head = newNode;
        else
            tail.setNext(newNode);

        tail = newNode;
        size++;
    }

    public Integer get(int index) {
        return findNodeByIndex(index).getData();
    }

    public boolean delete(int index) {
        if (findNodeByIndex(index) != null ) {

            if (index != 0)
                findNodeByIndex(index - 1).setNext(findNodeByIndex(index+1));
            else
                head = head.getNext();
        }
    }
}
```

```

        size--;
        return true;
    }
    return false;
}

public int size() {
    return size;
}

private Node findNodeByIndex(int index) {
    if (index < size && index >= 0) {
        Node curNode = head;
        int curIndex = 0;
        while (curIndex < index){
            if (curNode.getNext() != null) {
                curNode = curNode.getNext();
                curIndex++;
            }

        }
        return curNode;
    }
    else
        return null;
}
}

```

Колода карт:

```

package com.tasks3.carddeck; public class Deck { private Card[] deck; private int
                                currentCardNumber;

public Deck(){
    deck = new Card[Suit.values.length * Rank.values.length];
    currentCardNumber = deck.length - 1;
    int k = 0;
    for(int i = 0; i < Suit.values.length; i++){
        for(int j = 0; j < Rank.values.length; j++) {
            deck[k] = new Card(Rank.values[j], Suit.values[i]);
            k++;
        }
    }
}

```

```
}  
}
```

```
public void shuffle() {  
    if(currentCardNumber >= 0){  
        Card[] tempDeck = new Card[1];  
        for(int k = 0; k < currentCardNumber + 1; k++){  
            int a = random(currentCardNumber + 1);  
            tempDeck[0] = deck[k];  
            deck[k] = deck[a];  
            deck[a] = tempDeck[0];  
        }  
    }  
}
```

```
public void order() {  
    Card[] tempDeck = new Card[deck.length];  
    int n = 0;  
    for(int i = 0; i < Suit.values.length; i++) {  
        for(int j = 0; j < Rank.values.length; j++) {  
            for(int k = 0; k <= currentCardNumber; k++){  
                if(deck[k].getSuit().getName().equals(Suit.values[i].getName()) &&  
                    deck[k].getRank().getName().equals(Rank.values[j].getName())) {  
                    tempDeck[n] = deck[k];  
                    n++;  
                }  
            }  
        }  
    }  
    deck = tempDeck;  
}
```

```
public boolean hasNext() {  
    return currentCardNumber >= 0;  
}
```

```
public Card drawOne() {  
    if(hasNext()){  
        return deck[currentCardNumber--];  
    }  
    return null;  
}
```

```
public static int random(int deckLength){
```

```

    return (int)(Math.random() * deckLength);
}
}

```

Числа Фібоначчі:

```
package com.tasks3.fibonacci;
```

```
public class Fibonacci
```

```
{
```

```
    public long getNumber(int position){
```

```
        if ( position == 1 ) return 1;
```

```
        if(position <= 0) return -1;
```

```
        if (position == 2) return 1;
```

```
        long r = getNumber(position-2) + getNumber(position-1);
```

```
        if (r>0) return r;
```

```
        else return -1;
```

```
    }
```

```
    public Fibonacci(){
```

```
    }
```

```
}
```

