

Міністерство освіти, науки, молоді та спорту України
Національний університет «Львівська політехніка»

Кафедра СШІ

Лабораторна робота №3

Виконав:

ст. групи КН-107

Василик Ростислав

Прийняв:

викладач кафедри СШІ

Гасько Р.Т.

Львів-2017

Мета роботи: Створити класи для зберігання колоди, створити клас що описує зв'язний список та, використовуючи рекурсію, вивести на екран задане по порядковому номеру число Фібоначі.

Завдання 1

Код програми

```
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;
    }
}

}



---

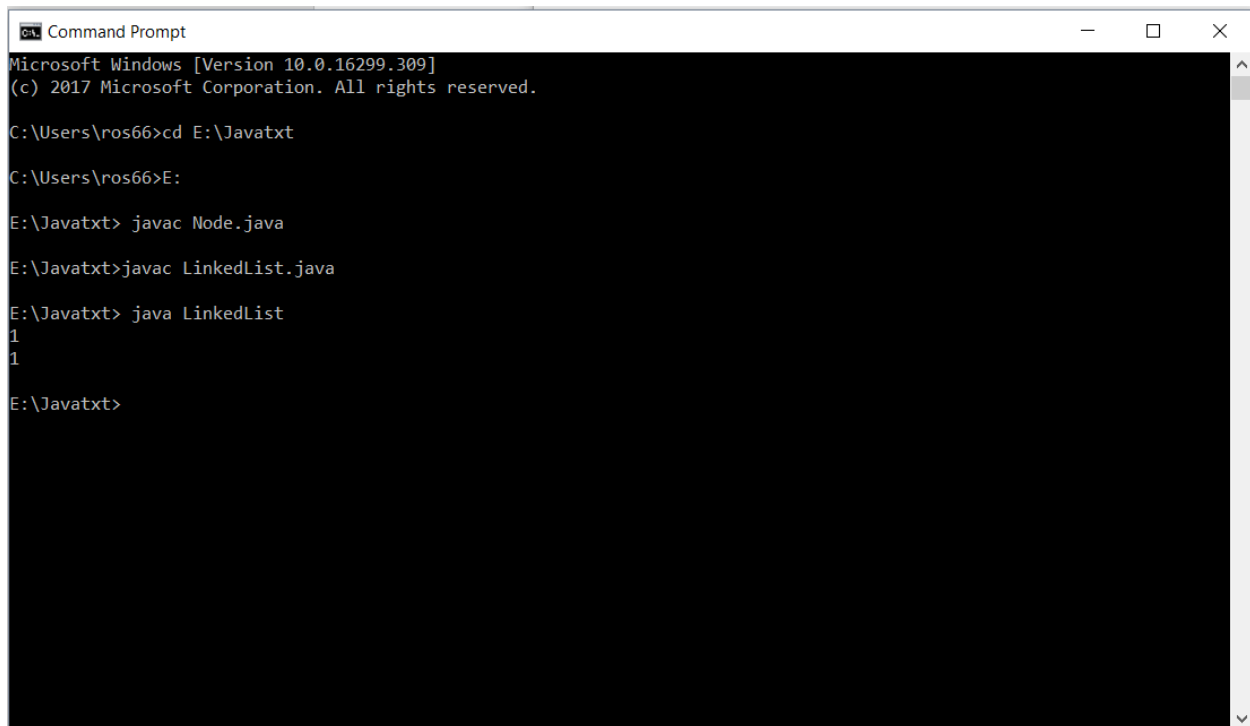

class Node{
    public Node next;
    public Integer data;

    public Node() {
    }

    public Node getNext() {
        return next;
    }
    public void setNext(Node next) {
        this.next = next;
    }
    public Integer getData() {
        return data;
    }
    public void setData(Integer data) {
        this.data = data;
    }
}

```

Вивід програми на консолі



```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ros66>cd E:\Javatxt

C:\Users\ros66>E:

E:\Javatxt> javac Node.java

E:\Javatxt>javac LinkedList.java

E:\Javatxt> java LinkedList
1
1

E:\Javatxt>
```

Завдання 2

Код програми:

```
public class Deck {
    private Card[] deck;
    private int currentCardNumber;

    public static void main(String[] args) {
        Deck deck1 = new Deck();
    }

    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]);
                System.out.print(deck[k].getSuit().getName() + " ");
                System.out.println(deck[k].getRank().getName());

                k++;
            }
            System.out.println();
        }
    }
}
```

```

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);
}
}

```

```

public class Card {
    private Rank rank;
    private Suit suit;

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

```

```

    }

    public Rank getRank() {
        return rank;
    }

    public void setRank(Rank rank) {
        this.rank = rank;
    }

    public Suit getSuit() {
        return suit;
    }

    public void setSuit(Suit suit) {
        this.suit = suit;
    }
}

```

```

public class Rank {
    public static final Rank ACE = new Rank("Ace");
    public static final Rank KING = new Rank("King");
    public static final Rank QUEEN = new Rank("Queen");
    public static final Rank JACK = new Rank("Jack");
    public static final Rank TEN = new Rank("10");
    public static final Rank NINE = new Rank("9");
    public static final Rank EIGHT = new Rank("8");
    public static final Rank SEVEN = new Rank("7");
    public static final Rank SIX = new Rank("6");

    public static Rank[] values = { ACE, KING, QUEEN, JACK, TEN, NINE, EIGHT,
SEVEN, SIX };

    private String name;

    Rank(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }
}

```

```

public class Suit {
    public static final Suit HEARTS = new Suit("HEARTS");
    public static final Suit DIAMONDS = new Suit("DIAMONDS");
    public static final Suit CLUBS = new Suit("CLUBS");
    public static final Suit SPADES = new Suit("SPADES");

    public static Suit[] values = { HEARTS, DIAMONDS, CLUBS, SPADES };
}

```

```

private String name;

Suit(String name) {
    this.name = name;
}

public String getName() {
    return name;
}
}

```

Вивід програми на консолі

```

Select Administrator Command Prompt
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32> cd E:\Javatxt

C:\WINDOWS\system32> E:

E:\Javatxt> javac Deck.java

E:\Javatxt> java Deck
HEARTS Ace
HEARTS King
HEARTS Queen
HEARTS Jack
HEARTS 10
HEARTS 9
HEARTS 8
HEARTS 7
HEARTS 6
DIAMONDS Ace
DIAMONDS King
DIAMONDS Queen
DIAMONDS Jack
DIAMONDS 10
DIAMONDS 9
DIAMONDS 8
DIAMONDS 7
DIAMONDS 6
CLUBS Ace
CLUBS King
CLUBS Queen
CLUBS Jack
CLUBS 10
CLUBS 9
CLUBS 8
CLUBS 7
CLUBS 6
SPADES Ace
SPADES King
SPADES Queen
SPADES Jack
SPADES 10
SPADES 9
SPADES 8
SPADES 7
SPADES 6

```

Завдання 3

Код програми:

```

public class Test
{
    public static int getNumber(int position){
        if ( position == 1 ) return 1;
        if(position <= 0) return -1;
        if (position == 2) return 1;

        int r = getNumber(position-2) + getNumber(position-1);
        if (r>0)

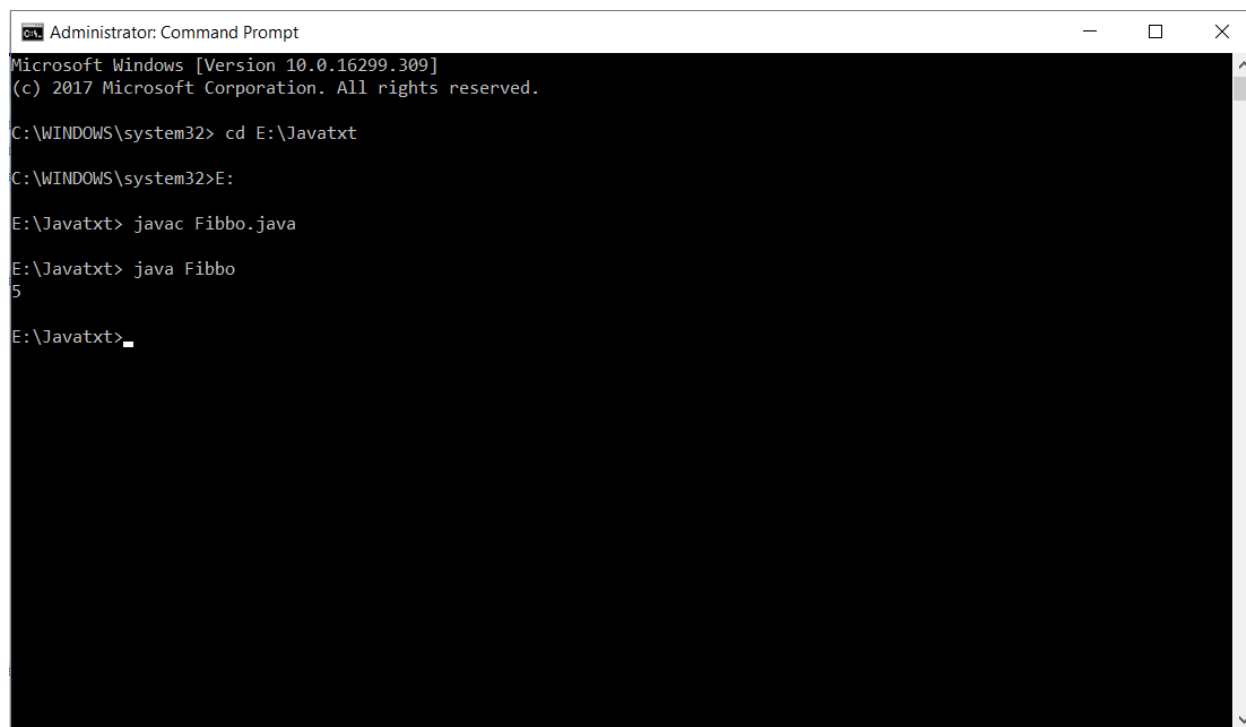
```

```

        return r;
        else return -1;
    }
    public static void main(String[] args)
    {
        int s=5;
        int p = getNumber(s);
        System.out.println(p);
    }
}

```

Вивід програми на консолі



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32> cd E:\Javatxt

C:\WINDOWS\system32>E:

E:\Javatxt> javac Fibbo.java

E:\Javatxt> java Fibbo
5

E:\Javatxt>

```

Висновок: На даній лабораторній роботі я навчився створювати класи для зберігання колоди, створювати клас що описує зв'язний список та, використовуючи рекурсію, виводити на екран задане по порядковому номеру число Фібоначі.