

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"  
Кафедра систем штучного інтелекту



## Лабораторна робота № 11

з дисципліни

«Об'єктно-орієнтоване програмування»

**Виконав:**

студент групи КН-109

Шмілик Т. О.

**Викладач:**

Гасько Р. Т.

Львів – 2018 р.

## Main.java

```
package
sample;

import java.util.Scanner;

public class Main {
    public static void main(String [] args){
        show();

        LinkedList LinkedList = new LinkedList();
        LinkedList LinkedList1 = new LinkedList();
        Scanner scan = new Scanner(System.in);
        String line = scan.nextLine();

        while(!("6".equals(line))) {
            //LinkedList LinkedList1 = new LinkedList();
            if ("1".equals(line)) {
                System.out.println("Please fill your personal card co-worker:");
                System.out.println("Station: ");
                String station = new String();
                station = scan.nextLine();
                System.out.println("Arrival: ");
                String arrival = new String();
                arrival = scan.nextLine();
                System.out.println("Departure: ");
                String departure = new String();
                departure = scan.nextLine();
                System.out.println("Free Seats");
                String fseats = new String();
                fseats = scan.nextLine();
                System.out.println("Number");
                String number = new String();
                number = scan.nextLine();
                System.out.println("Your ticket has been successfully created.");

                LinkedList.setT1(station);
                LinkedList.setT2(arrival);
                LinkedList.setT3(departure);
                LinkedList.setT4(fseats);
                LinkedList.setT5(number);
            }
        }
    }
}
```

```

        LinkedList1.add(LinkedList);

    }
    else if("2".equals(line)){
        System.out.println("Your index:");
        int b = scan.nextInt();
        System.out.println(LinkedList1.get(b));
    }

    else if("3".equals(line)){
        System.out.println("Your index:");
        int a = scan.nextInt();
        LinkedList1.delete(a);
    }
    else if("4".equals(line)){
        System.out.println(LinkedList1.size());
    }

    else if("5".equals(line)){
        System.out.println(LinkedList1.toString());
    }
    System.out.print("New command: ");
    line = scan.nextLine();
}

}

public static void show(){
    System.out.println("Help:\n" +
        "1 -> Fill in the ticket\n" +
        "2 -> Return ticket by index\n" +
        "3 -> Delete element by index\n" +
        "4 -> Return size\n" +
        "5 -> Return list with your tickets\n"+
        "6 -> Exit");
}
}

```

**LinkedList.java**

```
package  
sample;
```

```
public class LinkedList<T1, T2, T3, T4, T5> {
```

```
    public T1 getT1() {  
        return t1;  
    }
```

```
    public void setT1(T1 t1) {  
        this.t1 = t1;  
    }
```

```
    public T2 getT2() {  
        return t2;  
    }
```

```
    public void setT2(T2 t2) {  
        this.t2 = t2;  
    }
```

```
    public T3 getT3() {  
        return t3;  
    }
```

```
    public void setT3(T3 t3) {  
        this.t3 = t3;  
    }
```

```
    public T4 getT4() {  
        return t4;  
    }
```

```
    public void setT4(T4 t4) {  
        this.t4 = t4;  
    }
```

```
public T5 getT5() {  
    return t5;  
}
```

```
public void setT5(T5 t5) {  
    this.t5 = t5;  
}
```

```
private T1 t1;  
private T2 t2;  
private T3 t3;  
private T4 t4;  
private T5 t5;  
Node head;
```

```
public LinkedList() {  
  
}
```

```
/* Додати елемент в кінець списку */
```

```
public void add(LinkedList <T1, T2, T3, T4, T5> data) {  
  
    if (head == null) {  
        head = new Node();  
        head.setData(data);  
    } else {  
        Node node = new Node();  
        node.setData(data);  
        Node temp = head;  
        while (temp.getNext() != null) {  
            temp = temp.getNext();  
        }  
        temp.setNext(node);  
    }  
}
```

```
/* Отримати елемент по індексу, повертає null якщо такий елемент недоступний */
```

```
public LinkedList<T1, T2, T3, T4, T5> get(int index) {  
    if (index == 0) {  
        return head.getData();  
    }  
}
```

```

    } else if (index > 0) {
        if (head.getNext() != null) {
            Node temp = head;
            int position = 1;
            while (temp.getNext() != null) {
                if (position == index) {
                    return temp.getNext().getData();
                } else {
                    temp = temp.getNext();
                    position++;
                }
            }
        }
    }
    return null;
}

```

/\* Вилучення елемента за індексом, повертає true у разі успіху або false в іншому випадку \*/

```

public boolean delete(int index) {
    if (index == 0) {
        if (head == null) {
            return false;
        } else {
            if (head.getNext() == null) {
                head = null;
                return true;
            } else {
                head = head.getNext();
                return true;
            }
        }
    }
}

```

```

if (index > 0) {
    if (head.getNext() != null) {
        Node temp = head;
        int position = 1;
        while (temp.getNext() != null) {
            if (position == index) {
                temp.setNext(temp.getNext().getNext());
                return true;
            } else {
                temp = temp.getNext();
                position++;
            }
        }
    }
}

```

```

        }
    }
}
return false;
}

```

/\*Поверта розмір списку: якщо елементів в списку нема то повертає 0 (нуль)\*/

```

public int size() {
    int size = 0;
    if (head != null) {
        if (head.getNext() != null) {
            Node temp = head;
            size = 1;
            while (temp.getNext() != null) {
                temp = temp.getNext();
                size++;
            }
        } else {
            size = 1;
        }
    }
    return size;
}

```

@Override

```

public String toString() {
    StringBuffer str = new StringBuffer();

    Node toGet = head;
    while (toGet != null){
        str.append(toGet.getData().getT1().toString()).append(" ");
        str.append(toGet.getData().getT2().toString()).append(" ");
        str.append(toGet.getData().getT3().toString()).append(" ");
        str.append(toGet.getData().getT4().toString()).append(" ");
        str.append(toGet.getData().getT5().toString()).append(" ");
        toGet = toGet.getNext();
    }

    return str.toString();
}

```

```
public class Node {  
    private Node next;  
    private LinkedList <T1, T2, T3, T4, T5> data;  
  
    public Node() {  
    }  
  
    public Node getNext() {  
        return next;  
    }  
    public void setNext(Node next) {  
        this.next = next;  
    }  
    LinkedList <T1, T2, T3, T4, T5> getData() {  
        return data;  
    }  
    public void setData(LinkedList <T1, T2, T3, T4, T5> data) {  
        this.data = data;  
    }  
}  
}
```