

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

Кафедра систем штучного інтелекту

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

Виконала:
студентка групи КН-108
Бокшо Каріна
Викладач:
Гасько Р.Т.

Львів – 2017 р.

Завдання 1

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

```
public class LinkedList {  
    private Node tail;    private  
    Node head;
```

```
    private int size = 0;
```

```
    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;
    }

    public String toString() {
    StringBuilder resultString = new StringBuilder("");

    for (int i = 0; i < size; i++) {
    resultString.append(get(i));

    if (i < size-1) resultString.append(", ");
    }
    resultString.append("]");
    return resultString.toString();
    } }

```

Правильно

Результати тесту

		See full output
ВІРНО		
		See full output

ПЕРЕВІРКА

ЗБЕРЕГТИ

ПОКАЗАТИ ВІДПОВІДЬ

```

/* Вилучення елемента за індексом, повертає true у разі успіху або false в іншому випадку */
public boolean delete(int index) {}

/* Поверта розмір списку: якщо елементів в списку нема то повертає 0 (нуль) */
public int size() {}
}

```

ЗВЕРНІТЬ УВАГУ: елементи списку повинні мати номери починаючи з нуля!

```

1 package com.tasks3.linkedlist;
2 public class LinkedList {
3     private Node tail;
4     private Node head;
5     private int size = 0;
6     public LinkedList() {
7     }
8     public void add(Integer data) {
9         Node newNode = new Node();
10        newNode.setData(data);
11        if (size == 0)
12            head = newNode;
13        else
14            tail.setNext(newNode);
15        tail = newNode;
16        size++;

```

Правильно

Результати тесту

ВІРНО

[See full output](#)

[See full output](#)

ПЕРЕВІРКА

ЗБЕРЕГТИ

ПОКАЗАТИ ВІДПОВІДЬ

Ви використали 2 з 50 можливостей надіслати свої матеріали на розгляд.

Завдання 2

```
package com.tasks3.carddeck;
```

```
public class Deck {
```

```
    private static final int DECKSIZE = 36;
```

```
    private int decksize = 36;
```

```
    Card[] deck = new Card[DECKSIZE];
```

```
    public Deck(){
```

```
        for (int i = 0; i < 4; i++){
```

```
for (int j = 0; j < 9; j++){
```

```
    deck[i*9+j] = new Card(Rank.values[j], Suit.values[i]);
```

```
    }
```

```
}
```

```

    }

    public void shuffle() {
        Card temp = new Card(Rank.ACE, Suit.CLUBS);
        for (int i = 0; i < Math.random()*100; i++){
            int
randomNumberA = (int) (Math.random() * DECKSIZE);
            int
randomNumberB = (int) (Math.random() * DECKSIZE);
temp = deck[randomNumberA];
            deck[randomNumberA] = deck[randomNumberB];
            deck[randomNumberB] = temp;
        }
    }

    /* * Впорядкування колоди за мастями та значеннями
    * Порядок сотрування:
    * Спочатку всі карти з мастю HEARTS, потім DIAMONDS, CLUBS, SPADES
    * для кожної масті порядок наступний: Ace,King,Queen,Jack,10,9,8,7,6
    * Наприклад
    * HEARTS Ace
    * HEARTS King
    * HEARTS Queen
    * HEARTS Jack
    * HEARTS 10
    * HEARTS 9
    * HEARTS 8
    * HEARTS 7
    * HEARTS 6
    * І так далі для DIAMONDS, CLUBS, SPADES */
    public void order() {
        for (int i = 0; i < 4;
i++){
            for (int j = 0; j < 9; j++){
                deck[i*9+j] = new Card(Rank.values[j], Suit.values[i]);
            }
        }
    }

    //Повертає true у випадку коли в колоді ще доступні карти
    public boolean hasNext() {
        if (decksize > -1){
            return true;
        }
        else return false;
    }

    //"Виймає" одну карту з колоди, коли буде видано всі 36 карт повертає null
    //Карти виймаються з "вершини" колоди. Наприклад перший виклик видасть SPADES 6 потім
    //SPADES 7, ..., CLUBS 6, ..., CLUBS Ace і так далі до HEARTS Ace
    public Card drawOne() {
        decksize--;
        if (decksize > -1) {
            return deck[decksize];
        }
        else return null;
    }
}

```

Правильно

Результати тесту

		See full output
ВІРНО		
		See full output

ПЕРЕВІРКА

ЗБЕРЕГТИ

ПОКАЗАТИ ВІДПОВІДЬ

```
Suit(String name) {  
    this.name = name;  
}  
  
public String getName() {  
    return name;  
}  
}
```

```
1 package com.tasks3.carddeck;  
2 public class Deck {  
3     private static final int DECKSIZE = 36;  
4     private int decksize = 36;  
5     Card[] deck = new Card[DECKSIZE];  
6     public Deck(){  
7         for (int i = 0; i < 4; i++){  
8             for (int j = 0; j < 9; j++){  
9                 deck[i*9+j] = new Card(Rank.values[j], Suit.values[i]);  
10            }  
11        }  
12    }  
13    public void shuffle() {  
14        Card temp = new Card(Rank.ACE, Suit.CLUBS);  
15        for (int i = 0; i < Math.random()*100; i++){  
16            int randomNumberA = (int) (Math.random() * DECKSIZE);
```

Правильно

Результати тесту

ВІРНО

[See full output](#)

[See full output](#)

ПЕРЕВІРКА

ЗБЕРЕГТИ

ПОКАЗАТИ ВІДПОВІДЬ

Завдання 3

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

```
public class Fibonacci {
```

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

```
        if(position > 0 ) {
```

```
            if (position > 2)
```

```
                return getNumber(position - 1 ) + getNumber(position-2);
```

```
            else return 1;
```

```
        }    else
```

```
            return -1;
```

```
    }
```

Правильно

Результати тесту

ВІРНО

See full output

See full output

ПЕРЕВІРКА

ЗБЕРЕГТИ

ПОКАЗАТИ ВІДПОВІДЬ

► Тиждень 7

```
4 {
5     long first = 1;
6     long second = 1;
7     long result;
8     long iter = 2;
9
10    public long getNumber(int position){
11        if (position <= 30 && position > 0) {
12            if (position == 1) {
13                return first;
14            } else if (position == 2) {
15                return second;
16            } else if (iter != position) {
17                result = first + second;
18            }
19        }
20    }
```

Правильно

Результати тесту

ВІРНО

See full output

See full output

ПЕРЕВІРКА

ЗБЕРЕГТИ

ПОКАЗАТИ ВІДПОВІДЬ

Ви використали 8 з 50 можливостей надіслати свої матеріали на розгляд.