## 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 Гетьман Соломії

Викладач:

Гасько Р.Т.

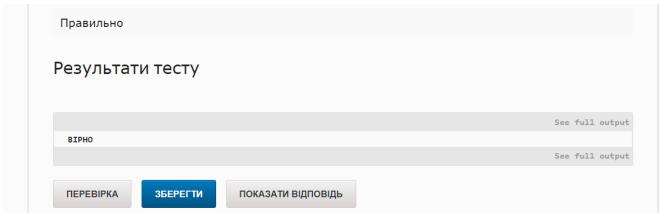
## Завдання 1

```
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));
         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();
}
</pre>
```



Завдання 2

```
package com.tasks3.carddeck;

public class Deck {

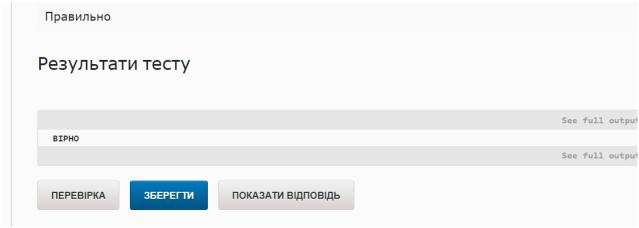
    private static final int DECKSIZE = 36;
    private int decksize = 36;

    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);
        }
}</pre>
```

```
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
   * I 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+i] = new Card(Rank.values[i], 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;
  }
```

}



Завданння 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 See full output ВІРНО See full output ПЕРЕВІРКА **ЗБЕРЕГТИ** ПОКАЗАТИ ВІДПОВІДЬ