Міністерство освіти і науки України

Національний університет „ Львівська політехніка “

# **ЗВІТ**

Про виконання лабораторної роботи №3

з об'єктно орієнтованого програмування

Виконала студентка групи КН-109

Романів Ярина

Керівник лабораторних занять:

Гасько Р.Т.

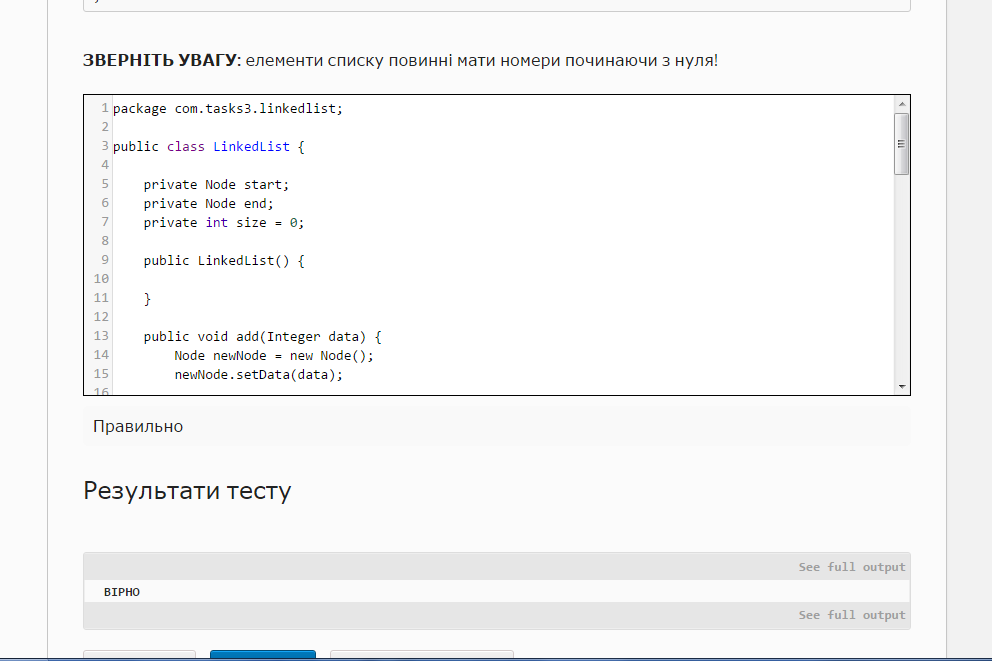
Львів – 2018

Завдання 1

Linked List

Код програми:

|  |
| --- |
| package com.tasks3.linkedlist; |
|  |  |
|  | public class LinkedList { |
|  |  |
|  | private Node start; |
|  | private Node end; |
|  | private int size = 0; |
|  |  |
|  | public LinkedList() { |
|  |  |
|  | } |
|  |  |
|  | public void add(Integer data) { |
|  | Node newNode = new Node(); |
|  | newNode.setData(data); |
|  |  |
|  | if (size == 0) |
|  | start = newNode; |
|  | else |
|  | end.setNext(newNode); |
|  |  |
|  | end = 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 |
|  | start = start.getNext(); |
|  |  |
|  | size--; |
|  | return true; |
|  | } |
|  | return false; |
|  | } |
|  |  |
|  | public int size() { |
|  | return size; |
|  | } |
|  |  |
|  | private Node findNodeByIndex(int index) { |
|  | if (index < size && index >= 0) { |
|  | Node curNode = start; |
|  | int curIndex = 0; |
|  | while (curIndex < index){ |
|  | if (curNode.getNext() != null) { |
|  | curNode = curNode.getNext(); |
|  | curIndex++; |
|  | } |
|  |  |
|  | } |
|  | return curNode; |
|  | } |
|  | else |
|  | return null; |
|  | } |
|  |  |
|  | } |



Завдання 2

Card Deck

Код програми:

package com.tasks3.carddeck;

public class Deck {

Card[] deck;

int index;

// Constructor

public Deck(){

index = 35;

int i = -1;

this.deck = new Card[Suit.values.length \* Rank.values.length];

for (Suit suit: Suit.values) {

for (Rank rank: Rank.values) {

deck[++i] = new Card(rank,suit);

}

}

}

//Перемішує колоду у випадковому порядку

public void shuffle() {

Card tmpCard;

int q = (int) (Math.random()\*300+700);

for (int n=0; n<q; n++){

int l = (int) (Math.random()\*32);

int m = (int) (Math.random()\*(36-l)+l);

tmpCard = this.deck[m];

this.deck[m] = this.deck[m-l];

this.deck[m-l] = tmpCard;

}

}

/\* \* 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() {

index = 35;

int i = -1;

this.deck = new Card[Suit.values.length \* Rank.values.length];

for (Suit suit: Suit.values) {

for (Rank rank: Rank.values) {

deck[++i] = new Card(rank,suit);

}

}

}

//Повертає true у випадку коли в колоді ще доступні карти

public boolean hasNext() {

return index > -1;

}

//"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() {

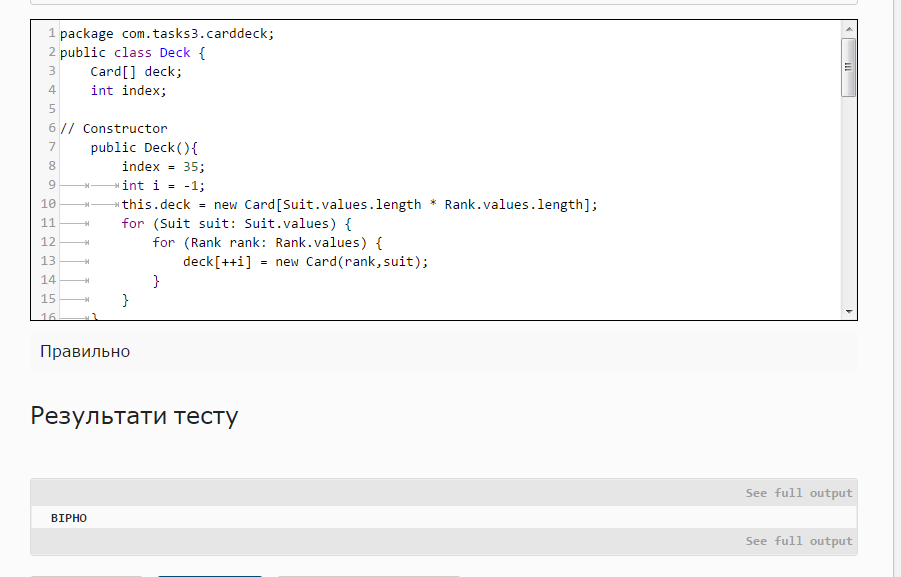
if (index >= 0)

return this.deck[index--];

else return null;

}

}



Завдання 3

Fibonacci

Код програми:

package com.tasks3.fibonacci;

public class Fibonacci {

public long getNumber(int position) {

if (position < 1) {

return -1L;

}

else if (position == 1 || position == 2) {

return 1L;

}

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

}

}

