МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій Кафедра систем штучного інтелекту



**Лабораторна робота №3**

# з дисципліни «OOП»

## Виконав:

Студент групи КН-109

Горблюк В. С.

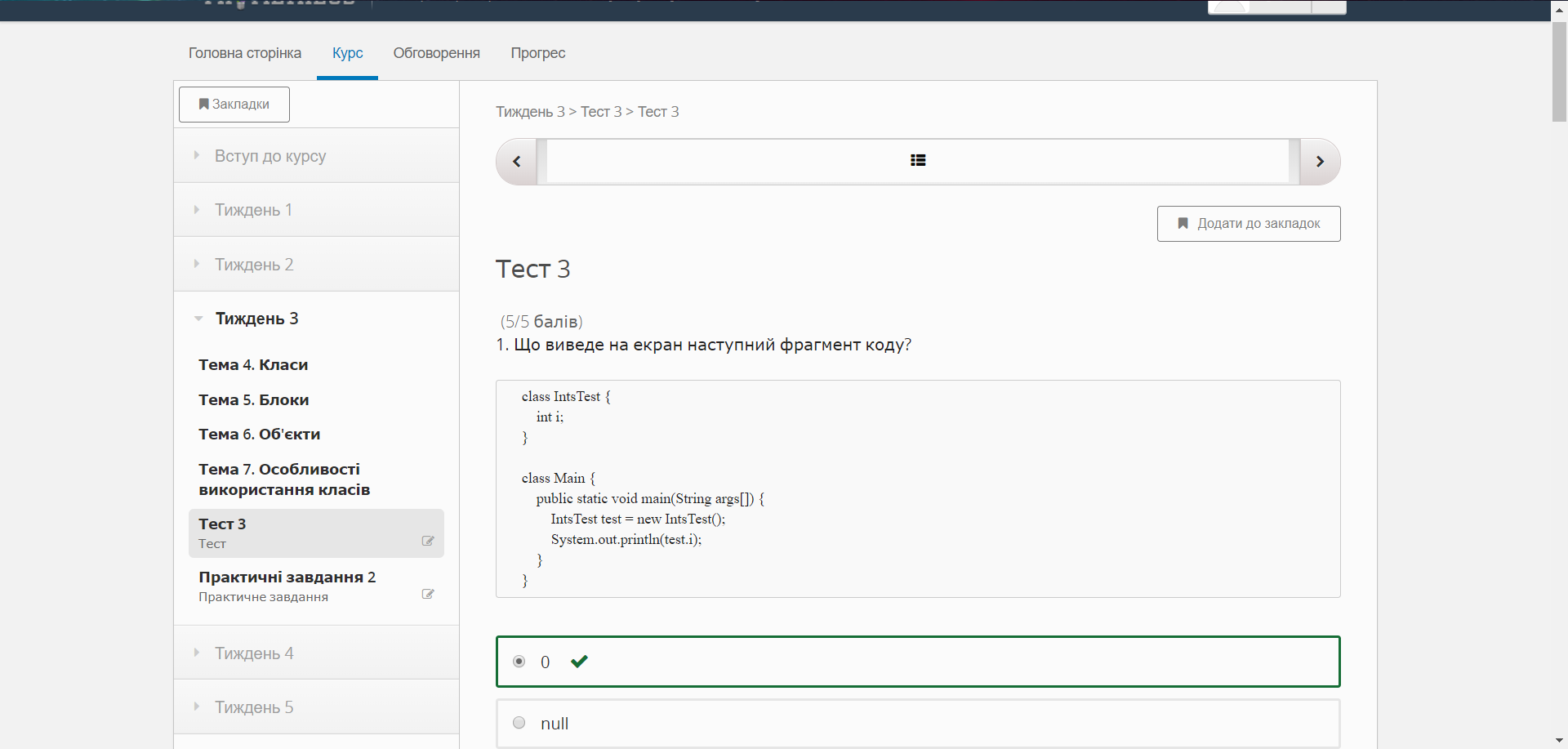
## Викладач:

Гасько Р. Т.

Львів – 2018

**Лабораторна робота №3**

# Успішно виконав тести третього тижня курсу «Основи програмування на Java».



1. Практичні 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();

}

}

2) 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();

}

}

3) package com.tasks3.fibonacci;

public class Fibonacci {

int[] fibonacciNumbers;

public long getNumberDynamic(int position){

fibonacciNumbers = new int [position];

for (int i = 0; i < position ; i++) {

if (i > 1)

fibonacciNumbers[i] = fibonacciNumbers[i-1] + fibonacciNumbers[i-2];

else

fibonacciNumbers[i] = 1;

}

return fibonacciNumbers[position-1];

}

public int getNumber(int position) {

if (position > 2)

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

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

return 1;

else

return -1;

}

}