

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Лабораторна робота №1

з дисципліни «ООП»

Виконав:

Студент групи КН-109

Савчин В. А.

Викладач:

Гасько Р. Т.

Лабораторна робота №1

1) Я успішно пройшов тести із першого тижня курсу «Основи програмування на Java».

The screenshot shows a web interface for a course. On the left is a sidebar with a navigation menu. The main area displays the results of 'Тест 3' (Test 3), which is marked as completed with a green checkmark. The test score is 5/5 points. The question asks what will be printed on the screen for a given Java code snippet. The code defines a class 'IntsTest' with an 'int i' variable and a 'Main' class with a 'main' method that creates an 'IntsTest' object and prints its 'i' value. The answer options are '0', 'null', and 'Помилка компіляції' (Compilation error). The option '0' is selected and marked correct.

Тиждень 1

Тиждень 2

Тиждень 3

Тема 4. Класи

Тема 5. Блоки

Тема 6. Об'єкти

Тема 7. Особливості використання класів

Тест 3

Тест

Практичні завдання 2

Практичне завдання

Тиждень 4

Тиждень 5

Тиждень 6

Додати до закладок

Тест 3

(5/5 балів)

1. Що виведе на екран наступний фрагмент коду?

```
class IntsTest {
    int i;
}

class Main {
    public static void main(String args[]) {
        IntsTest test = new IntsTest();
        System.out.println(test.i);
    }
}
```

☒ 0 ✓

☐ null

☐ Помилка компіляції

2) Практичні завдання:

1) package com.tasks3.linkedlist;

```
public class LinkedList {
```

```
    int size;
```

```
    Node head;
```

```
    Node tail;
```

```
    Node current;
```

```
    public LinkedList() {
```

```
        size = 0;
```

```
        head=null;
```

```
        tail=null;
```

```
    }
```

```
    public void add(Integer data) {
```

```

Node new_Node=new Node();

new_Node.setData(data);

if(tail==null) {

    head=new_Node;

    tail=new_Node;

}else {

tail.setNext(new_Node);

tail=new_Node;

}

size++;

}

public Integer get(int index) {

    current=head;

    if(index>size) {

        return null;

    }

    else if(index==0){

        {

            return head.getData();

        }

    }

    else {    while(index>0) {

        if(current==null) {

            return null;

        }else {

            current=current.getNext();

            index--;

        }

    }

    return current.getData();

```

```

    }

}

public boolean delete(int index) {

    if (size() == 0 || index > size()-1) {

        return false;

    }

    if(index<0) {

        size--;

        return false;

    }

    if(index==0) {

        head=head.getNext();

        size--;

        return true;

    }

    else if ((index == 0) && (head == tail)) {

        head = head.getNext();

        if (head == null) {

            tail = null;

        }

        size--;

        return true;

    }

    else {

        Node previous=head;

        Node current=head.getNext();

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

```

```

        if(i+1 == index) {

            previous.setNext(null);

            previous.setNext(current.getNext());

            current=null;

            i=size;

        }else {

            previous=current;

            current=current.getNext();

        }

    }

    size--;

    return true;

}

}

```

```

public int size() {

    return size;

}

```

```

class Node{

    private Integer data;

    private Node next;

    public Node() {

    }

    public Node getNext() {

        return next;

    }

    public void setNext(Node next) {

        this.next = next;

    }

    public Integer getData() {

```

```

        return data;
    }

    public void setData(Integer data) {

        this.data = data;

    }

}

```

2) package com.tasks3.carddeck;

```

public class Deck {

    private Card cards[];

    private int size;

    private Card used[];

    public Deck()

    {

        int index = 0;

        this.size = 36;

        cards = new Card[this.size];

        used = new Card[this.size];

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

        {

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

            {

                this.cards[index++] = new Card(Rank.values[j], Suit.values[i]);

            }

        }

    }

    public void shuffle() {

```

```

for(int i = 0; i < this.size; i++) {

    int random_num = (int)(Math.random()*this.size);

    Card tmp = this.cards[i];

    this.cards[i] = this.cards[random_num];

    this.cards[random_num] = tmp;

}

}

public void order() {

    int index = 0;

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

    {

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

        {

            boolean create = true;

            for(int k = size; k < 36; k++)

            {

                if((used[k].getRank() == Rank.values[i])&&(used[k].getSuit() == Suit.values[j]))

                    create = false;

            }

            if(create) {

                this.cards[index].setRank(Rank.values[j]);

                this.cards[index].setSuit(Suit.values[i]);

                index++;

            }

        }

    }

}

public boolean hasNext() {

    if(size != 0)

```

```

        return true;

    else

        return false;

    }

    public Card drawOne() {

        size--;

        if(size >= 0) {

            used[size] = new Card(cards[size].getRank(), cards[size].getSuit());

            return cards[size];

        }

        else

            return null;

    }

}

```

3) package com.tasks3.fibonacci;

```

public class Fibonacci

{

    public long getNumber(int position){

        if(position%1 != 0 || position<1) {

            return -1;

        }

        else if(position <3){

            return 1;

        }else{

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

        }

    }

}

```