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

Семкович В.І.

## Викладач:

Гасько Р. Т.

Львів – 2018

**Лабораторна робота №3**

# Успішно виконав тести 3 тижня курсу «Основи програмування на Java».



2) Практичні завдання

1.

package com.tasks3.linkedlist;

public class LinkedList {

private Node finish;

private Node start;

private int size = 0;

public LinkedList() {

}

public void add(Integer data) {

Node newNode = new Node();

newNode.setData(data);

if (size == 0){

start = newNode;

}

else{

finish.setNext(newNode);

}

finish = newNode;

size++;

}

public Integer get(int index) {

return nodeSearch(index).getData();

}

public boolean delete(int index) {

if (nodeSearch(index) != null ) {

if (index == 0) {

start = start.getNext();

}

else{

nodeSearch(index - 1).setNext(nodeSearch(index+1));

}

size--;

return true;

}

return false;

}

public int size() {

return size;

}

2.

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);

}

}

}

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;

}

}

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);

}

}

}

public boolean hasNext() {

return index > -1;

}

public Card drawOne() {

if (index >= 0)

return this.deck[index--];

else return null;

}

public void deckPrn(){

System.out.println("---Cards in deck-"+(index+1)+"---");

for (int j=0; j<index+1; j++){

System.out.print("|");

System.out.print(this.deck[j].getSuit().getName());

System.out.print(" ");

System.out.println(this.deck[j].getRank().getName());

}

}

public void cardPrn(Card crd){

if (crd != null){

System.out.print(crd.getSuit().getName());

System.out.print(" ");

System.out.println(crd.getRank().getName());

}

else System.out.println("Card is NULL");

}

public static void main(String[] args) {

Deck dk1 = new Deck();

dk1.shuffle();

dk1.deckPrn();

//dk1.order();

//dk1.cardPrn(dk1.deck[10]);

/\*

for (int t=0; t<36; t++){

dk1.cardPrn(dk1.drawOne());

System.out.println("----------------------");

}

System.out.println(dk1.hasNext());

dk1.deckPrn();

\*/

}

private Node nodeSearch(int index) {

if (index < size && index >= 0) {

Node currentNode = start;

int currentIndex = 0;

while (currentIndex < index){

if (currentNode.getNext() != null) {

currentNode = currentNode.getNext();

currentIndex++;

}

}

return currentNode;

}

else

return null;

}

}

3.

package com.tasks3.fibonacci;

import java.util.Scanner;

public class Fibonacci {

// boolean change=false;

// public long getNumber(int position) {

// if (!change) {

// if(position<= 0)

// return -1;

// position--;

// change = true;

// }

// if (position == 1 || position == 0)

// return 1;

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

// }

public long getNumber(int position) {

if(position< 1)

return -1;

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

return 1;

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

}

public static void main(String arg[]) {

Fibonacci f = new Fibonacci();

Scanner scanner = new Scanner(System.in);

System.out.println(f.getNumber(scanner.nextInt()));

}

}