# Міністерство освіти, науки, молоді та спорту України Національний університет «Львівська політехніка»

**Кафедра СШІ**

Лабораторна робота №4 З ООП

# Виконав:

ст. групи КН-107 Тимків А.І. Прийняв: Старший викладач СШІ

Гасько Р.Т.

Львів-2018

Завдання 1 Використовуючи наведені нижче класи змініть порядок сортування на зворотній.

Код:

**public class Sort {**

**private static void sort(int[] array, Comparator comp) {**

**for (int gap = array.length / 2; gap > 0; gap /= 2) {**

**for (int i = gap; i < array.length; i++) {**

**int val = array[i];**

**int j;**

**for (j = i; j >= gap && comp.compare(array[j - gap], val) > 0; j -= gap) {**

**array[j] = array[j - gap];**

**}**

**array[j] = val;**

**}**

**}**

**}**

**public static void main(String[] args){**

**int[] array = {1,5,2,4,10,6,0,3,10};**

**Comparator comp = new Comparator();**

**//Change your code here**

**sort(array, comp);**

**//Change your code here**

**int[] array2 = new int[array.length];**

**int k = 0;**

**for (int i = array.length-1; i >= 0; i--,k++) {**

**array2[k] = array[i];**

**}**

**for (int i = 0; i < array2.length; i++) {**

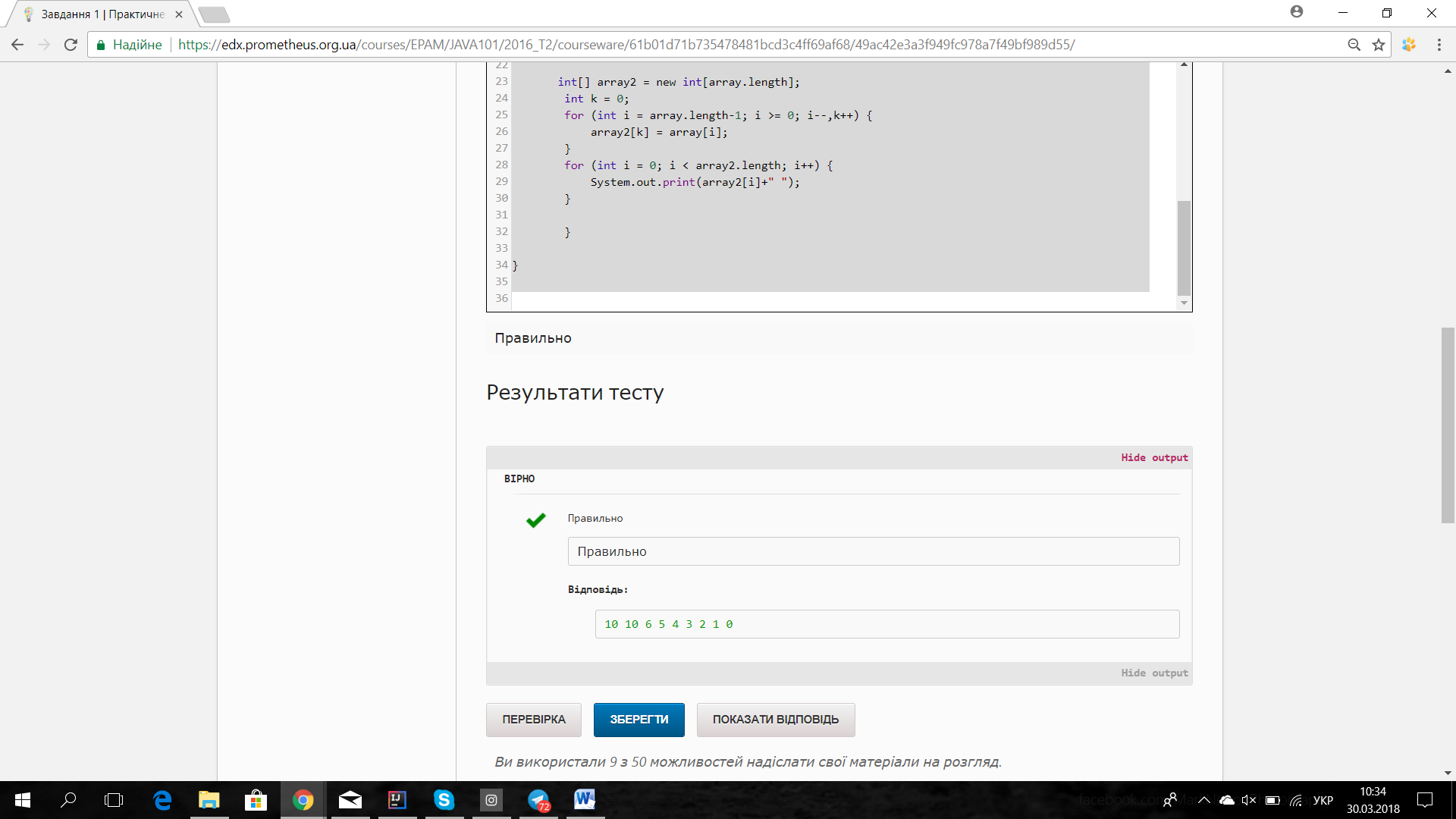
**System.out.print(array2[i]+" ");**

**}**

**}**

**}**

Результат:



for(int j = 0; j < Rank.values.length; j++) {

deck[k] = new Card(Rank.values[j], Suit.values[i]);

k++;

}

}

}

public void shuffle() {

if(currentCardNumber >= 0){

Card[] tempDeck = new Card[1];

for(int k = 0; k < currentCardNumber + 1; k++){

int a = random(currentCardNumber + 1);

tempDeck[0] = deck[k];

deck[k] = deck[a];

deck[a] = tempDeck[0];

}

}

}

public void order() {

Card[] tempDeck = new Card[deck.length];

int n = 0;

for(int i = 0; i < Suit.values.length; i++) {

for(int j = 0; j < Rank.values.length; j++) {

for(int k = 0; k <= currentCardNumber; k++){

if(deck[k].getSuit().getName().equals(Suit.values[i].getName()) && deck[k].getRank().getName().equals(Rank.values[j].getName())) {

tempDeck[n] = deck[k];

n++;

}

}

}

}

deck = tempDeck;

}

public boolean hasNext() {

return currentCardNumber >= 0;

}

public Card drawOne() {

if(hasNext()){

return deck[currentCardNumber--];

}

return null;

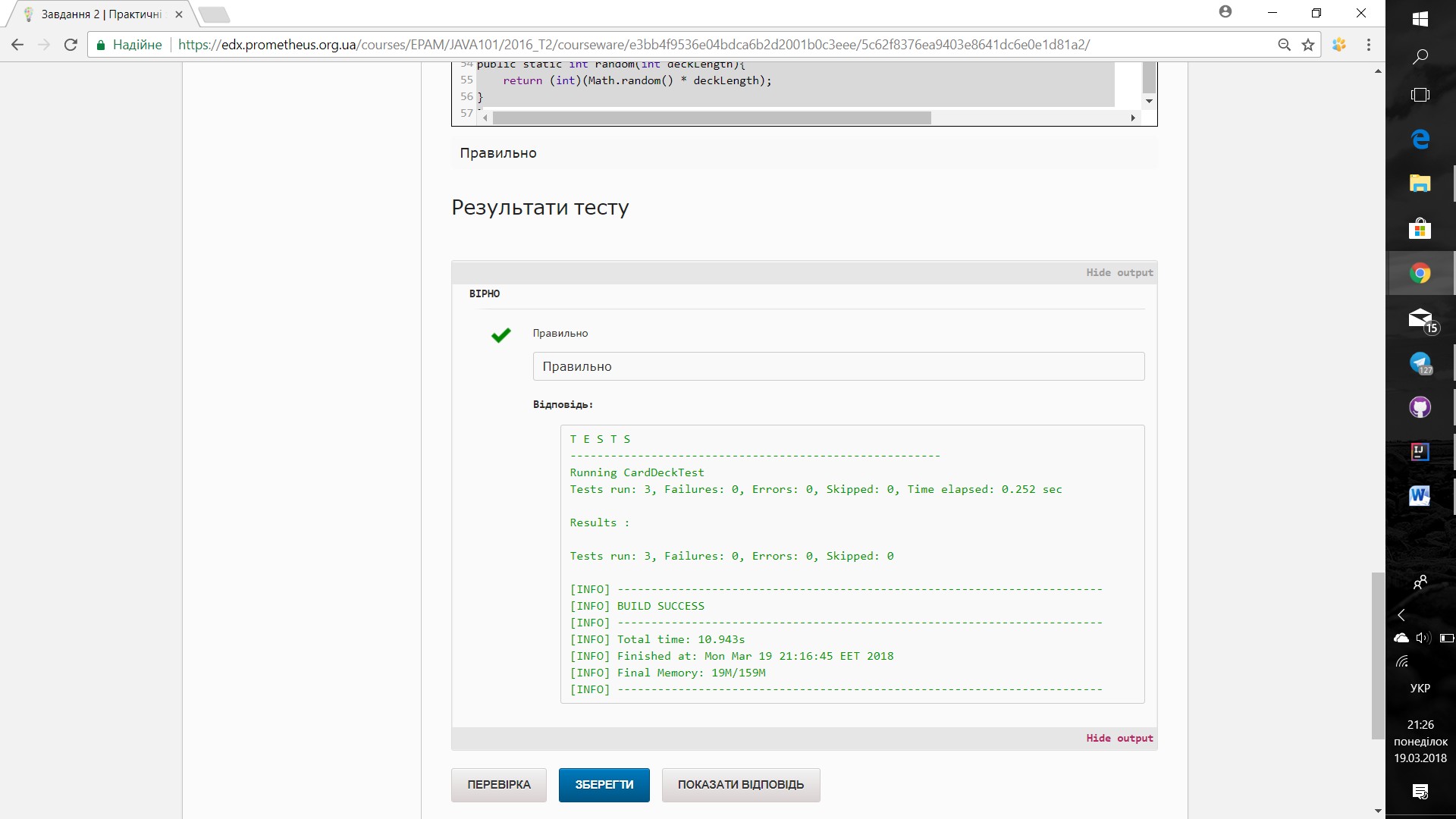
}

public static int random(int deckLength){

return (int)(Math.random() \* deckLength);

}

}



Результат:

Завдання 3 Використовуючи рекурсію, виведіть на екран задане по порядковому номеру число Фібоначі.

package com.tasks3.fibonacci;

public class Fibonacci { public Fibonacci(){}

public int getNumber(int position) { if(position > 0 && position < 93) {

if (position > 2)

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

else

return 1;

}

else

return -1;

}

}

Результат:

