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

**Кафедра СШІ**

Лабораторна робота №5 З ООП

# Виконав:

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

Гасько Р.Т.

Львів-2018

Завдання 1 Створіть застосування, що реалізуватиме патерн [Команда](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%B0_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))

Реалізуйте інтерфейс Command

Код:

**package com.tasks5.command;**

**public class Application {**

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

**//YOUR CODE COMES HERE**

**class Commands{**

**void help(){**

**System.out.println("Help executed");**

**}**

**void echo(String s){**

**System.out.println(s);**

**}**

**void date(){**

**System.out.println(System.currentTimeMillis());**

**}**

**void exit(){**

**System.out.println("Goodbye!");**

**}**

**}**

**// ConcrateCommand**

**class HelpCommand implements Command{**

**private Commands com;**

**HelpCommand(Commands com){**

**this.com = com;**

**}**

**@Override**

**public void execute() {**

**com.help();**

**}**

**}**

**// ConcrateCommand**

**class EchoCommand implements Command{**

**private Commands com;**

**String s;**

**EchoCommand(Commands com, String s){**

**this.com = com;**

**this.s = s;**

**}**

**@Override**

**public void execute() {**

**com.echo(s);**

**}**

**}**

**// ConcrateCommand**

**class DateCommand implements Command{**

**private Commands com;**

**DateCommand(Commands com){**

**this.com = com;**

**}**

**@Override**

**public void execute() {**

**com.date();**

**}**

**}**

**// ConcrateCommand**

**class ExitCommand implements Command{**

**private Commands com;**

**ExitCommand(Commands com){**

**this.com = com;**

**}**

**@Override**

**public void execute() {**

**com.exit();**

**}**

**}**

**if (args != null && args.length > 0 && args.length < 3) {**

**Commands commands = new Commands();**

**switch (args[0]){**

**case "help":**

**if (args.length == 1){**

**Command help = new HelpCommand(commands);**

**help.execute();**

**}**

**else**

**System.out.println("Error");**

**break;**

**case "echo":**

**Command echo = new EchoCommand(commands,args[1]);**

**echo.execute();**

**break;**

**case "date":**

**if (args[1] == "now"){**

**Command date = new DateCommand(commands);**

**date.execute();**

**}**

**else**

**System.out.println("Error");**

**break;**

**case "exit":**

**if (args.length == 1) {**

**Command exit = new ExitCommand(commands);**

**exit.execute();**

**}**

**else**

**System.out.println("Error");**

**break;**

**default:**

**System.out.println("Error");**

**}**

**}**

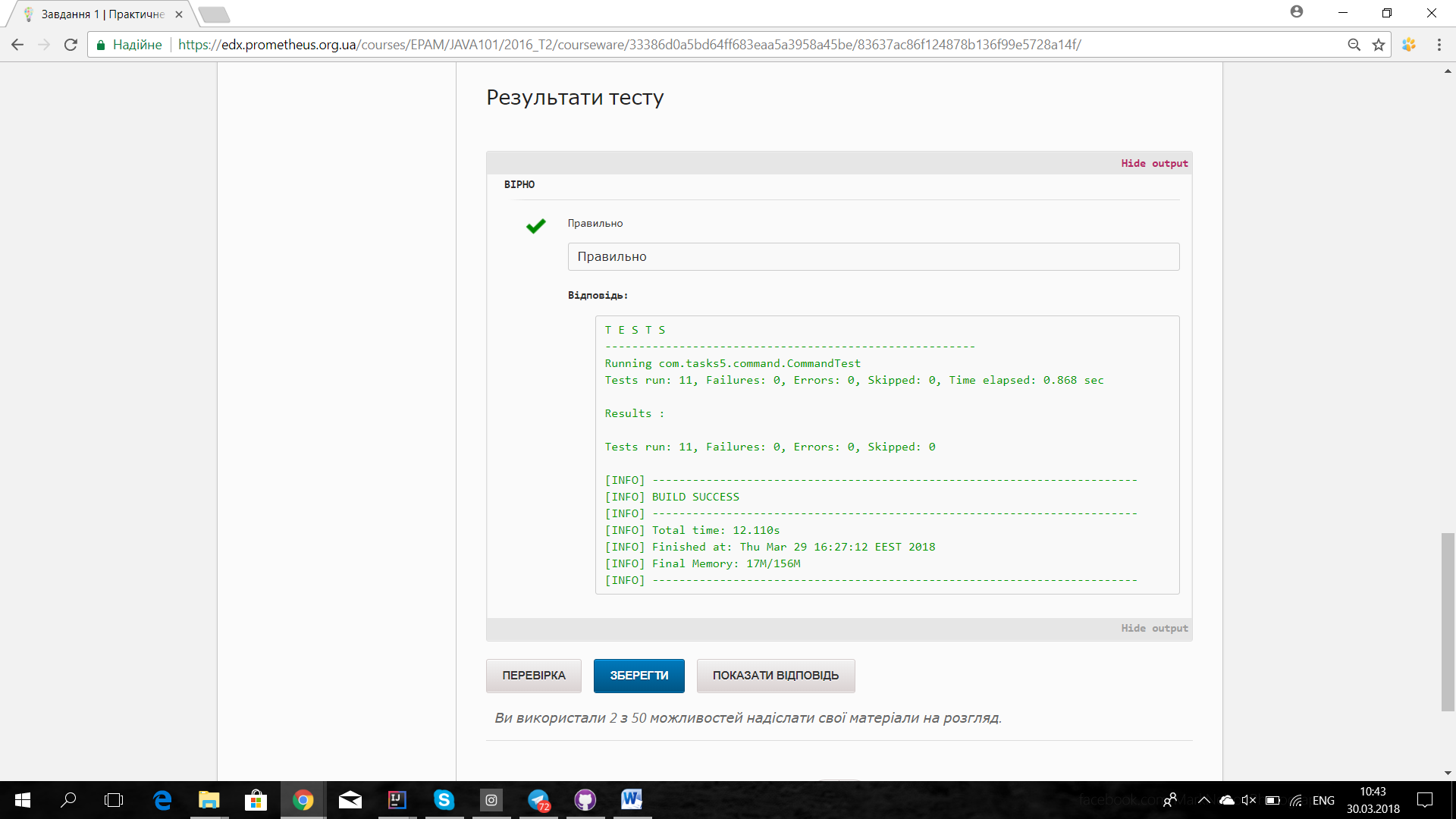
**else {**

**System.out.println("Error");**

**}**

**}**

**}**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;

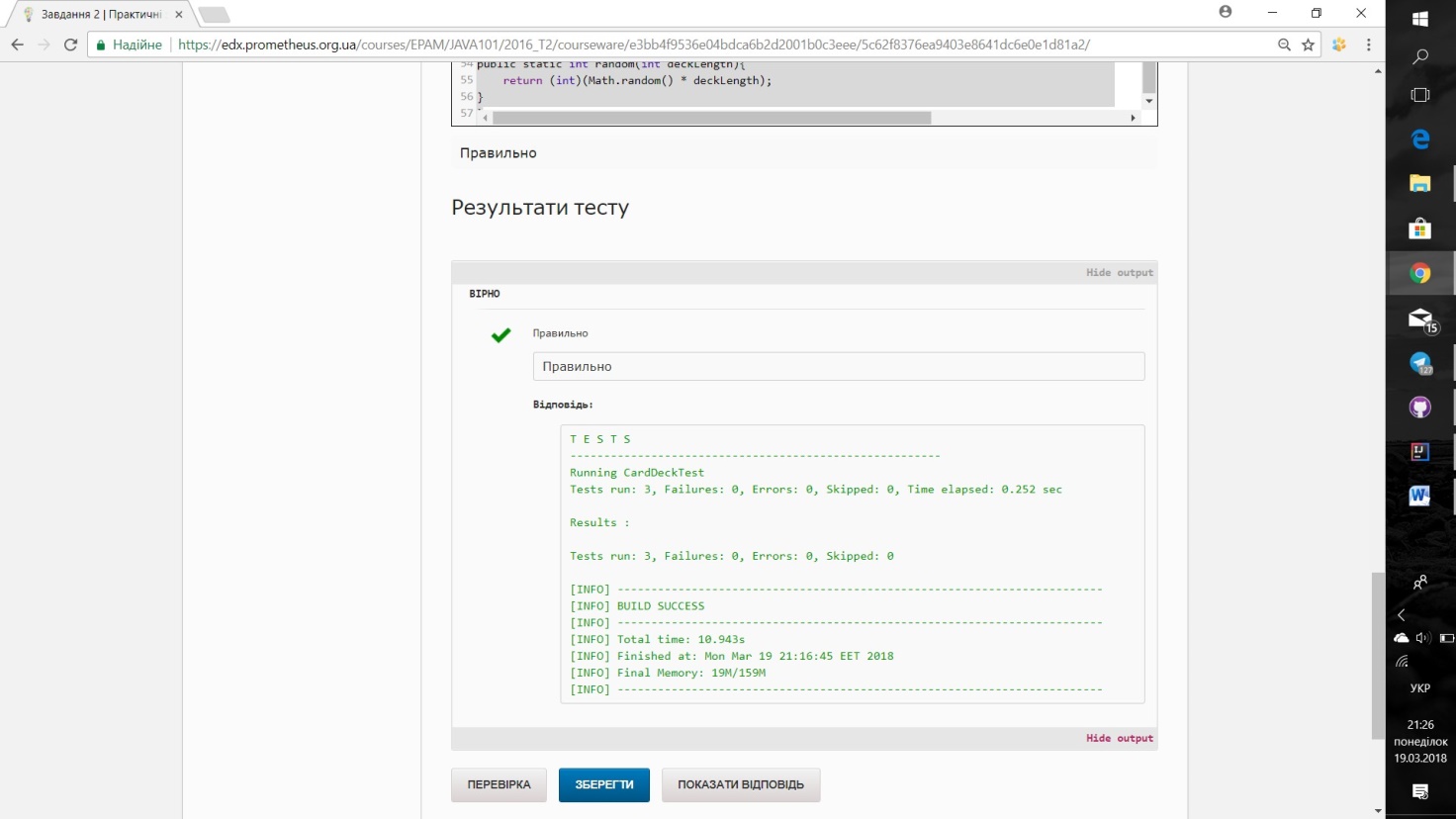
}

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:

