

**Лабораторна робота № 2**  
з дисципліни  
«Об’єктно-орієнтоване програмування»

**Виконала:**  
студентка групи КН-109  
Пелешак Ю. М.  
**Викладач:**  
Гасько Р.Т.

Львів – 2018 р.

# 1. Створити застосування, що знаходить корені рівняння виду : $ax^2 + bx + c = 0$ Коефіцієнти задаються через змінні.

```
public class SquareRoot {
    public static void main(String[] args) {

//задаємо змінні

        double a = 3;
        double b = 2.5;
        double c = -0.5;

        double x_1, x_2, d;

//перевіряємо чи дискримінант рівня >= 0 та чи коефіцієнт a!=0

        if (b*b - 4*a*c>=0 & a!=0 ) {

//шукаємо корені за формулою

            d =Math.sqrt(b*b - (4*a*c));
            x_1 = ((-b) + d)/(2*a);
            x_2 = ((-b) - d)/(2*a);
            System.out.println("x1="+ x_1);
            System.out.println("x2="+ x_2);
        }

//якщо дискримінант менший від нуля або змінні a та b = 0, то просто виводимо
x1= , x2=

        else if(a==0 & b==0 | b*b - 4*a*c<0 ) {
            System.out.println("x1=");
            System.out.println("x2=");
        }

//в іншому випадку виводимо, x1=0, x2=0
        else{
            System.out.println("x1="+0.0);
            System.out.println("x2="+0.0);
        }
    }
}
```

## Програму зараховано на Прометеусі

The screenshot shows the IntelliJ IDEA IDE with a project named 'SquareRoot'. The main class is 'SquareRoot.java', which contains the following code:

```
1 public class SquareRoot {
2     public static void main(String[] args) {
3         double a = 3;
4         double b = 2.5;
5         double c = -0.5;
6
7         double x_1, x_2, d;
8
9         if (b*b - 4*a*c >= 0 & a != 0) {
10             d = Math.sqrt(b*b - (4*a*c));
11             x_1 = ((-b) + d) / (2*a);
12             x_2 = ((-b) - d) / (2*a);
13             System.out.println("x1=" + x_1);
14             System.out.println("x2=" + x_2);
15         }
16         else if (a==0 & b==0 | b*b - 4*a*c < 0) {
17             System.out.println("x1=");
18             System.out.println("x2=");
19         }
20         else {
21             System.out.println("x1=" + 0.0);
22         }
23     }
24 }
```

The Run window shows the output:

```
x1=0.16666666666666666
x2=-1.0
Process finished with exit code 0
```

Below the IDE, a browser window shows the same code on the Prometheus website. The website displays the code with line numbers and a 'Відрізки' (Segments) button. The output section shows 'Правильно' (Correct) and 'Результати тесту' (Test results). The test results section shows 'ВІРНО' (Correct) and a 'ЗБЕРЕГТИ' (Save) button.

## 2) Використовуючи цикл for виведіть на екран матрицю:

```
* 2 3 4 *
6 * 8 * 10
11 12 * 14 15
16 * 18 * 20
* 22 23 24 *
```

```
public class MatrixPrint {
    public static void main(String args[]) {
        int size = 5;

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

```

        for (int j = 0; j < size; j++)
        {
//рахуємо число

            int num = size * i + j + 1;

//виводимо «*» на діагоналях матриці

            if (j == i || j == size - 1 - i)
            {
                System.out.print(' ');
                System.out.print('*');
            }

//виводимо числа, які стоять не на діагоналях;
Якщо число менше 10, то потрібно зробити додатковий відступ

            else
            {
                if (num < 10)
                    System.out.print(' ');
                System.out.print(num);
            }

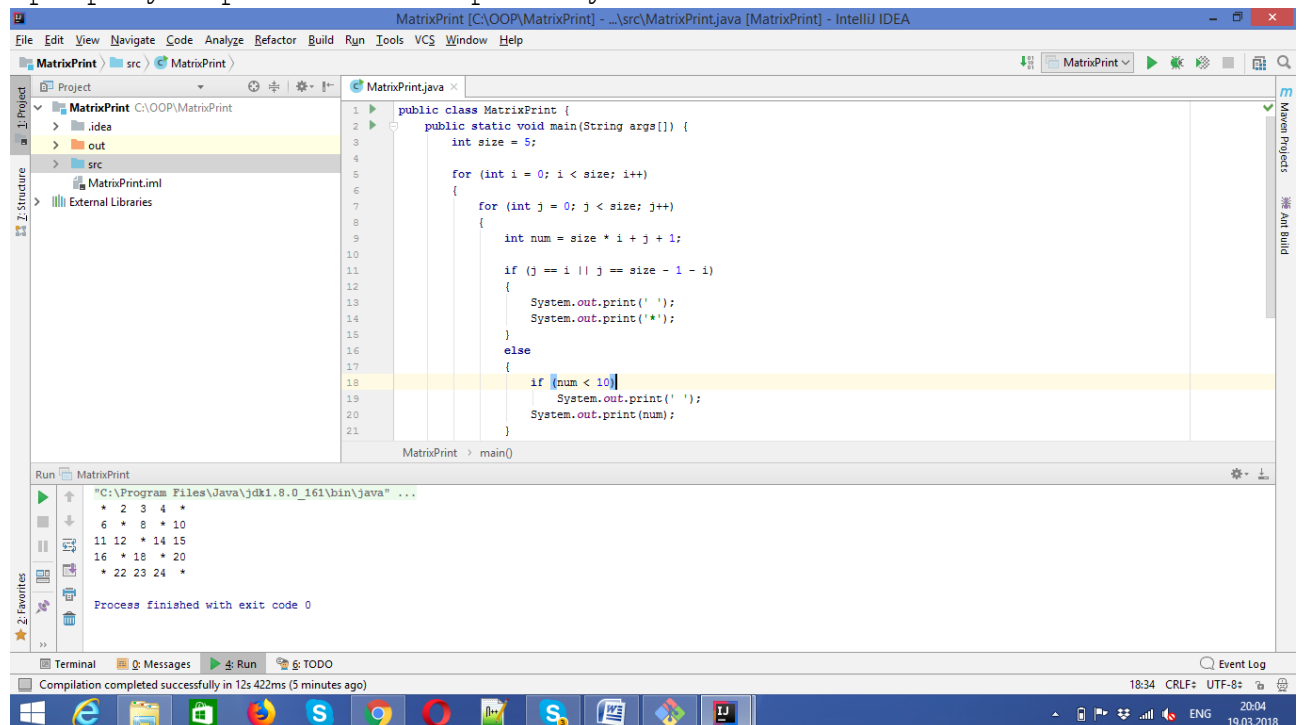
            System.out.print(' ');
        }

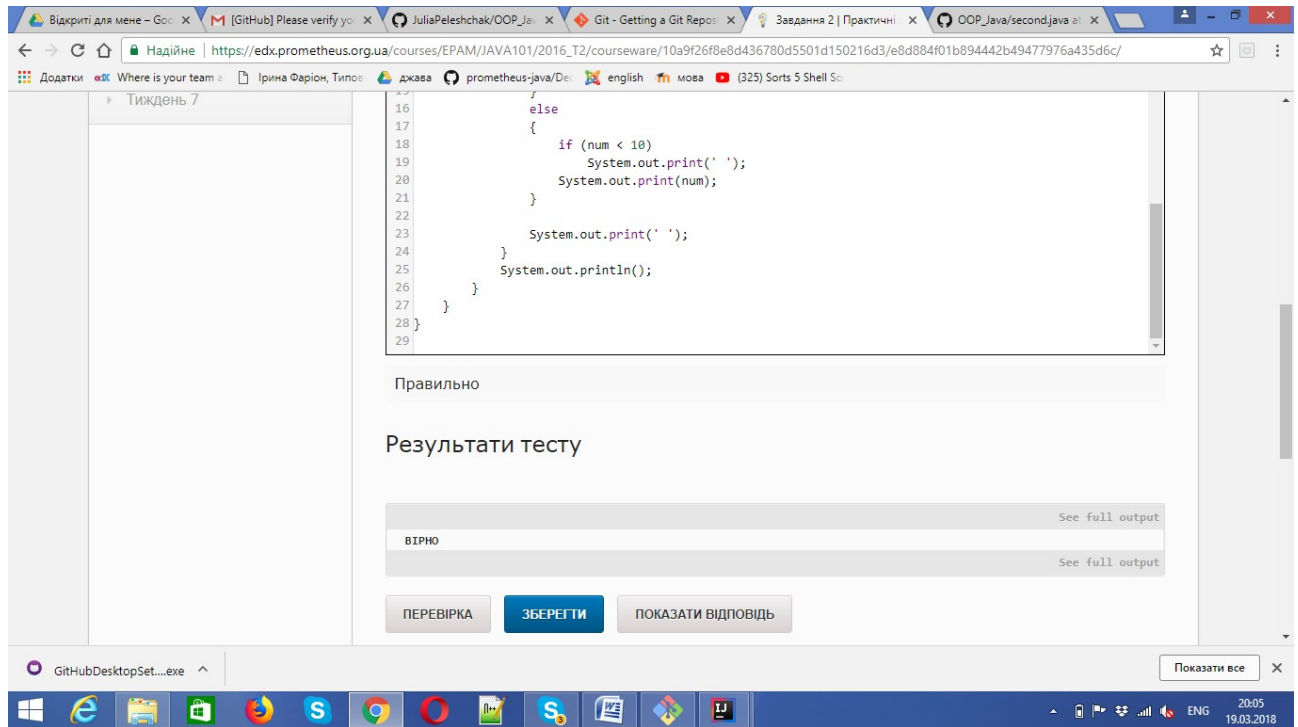
//переходимо на новий рядок

        System.out.println();
    }
}

```

Програму зараховано на Прометеусі.

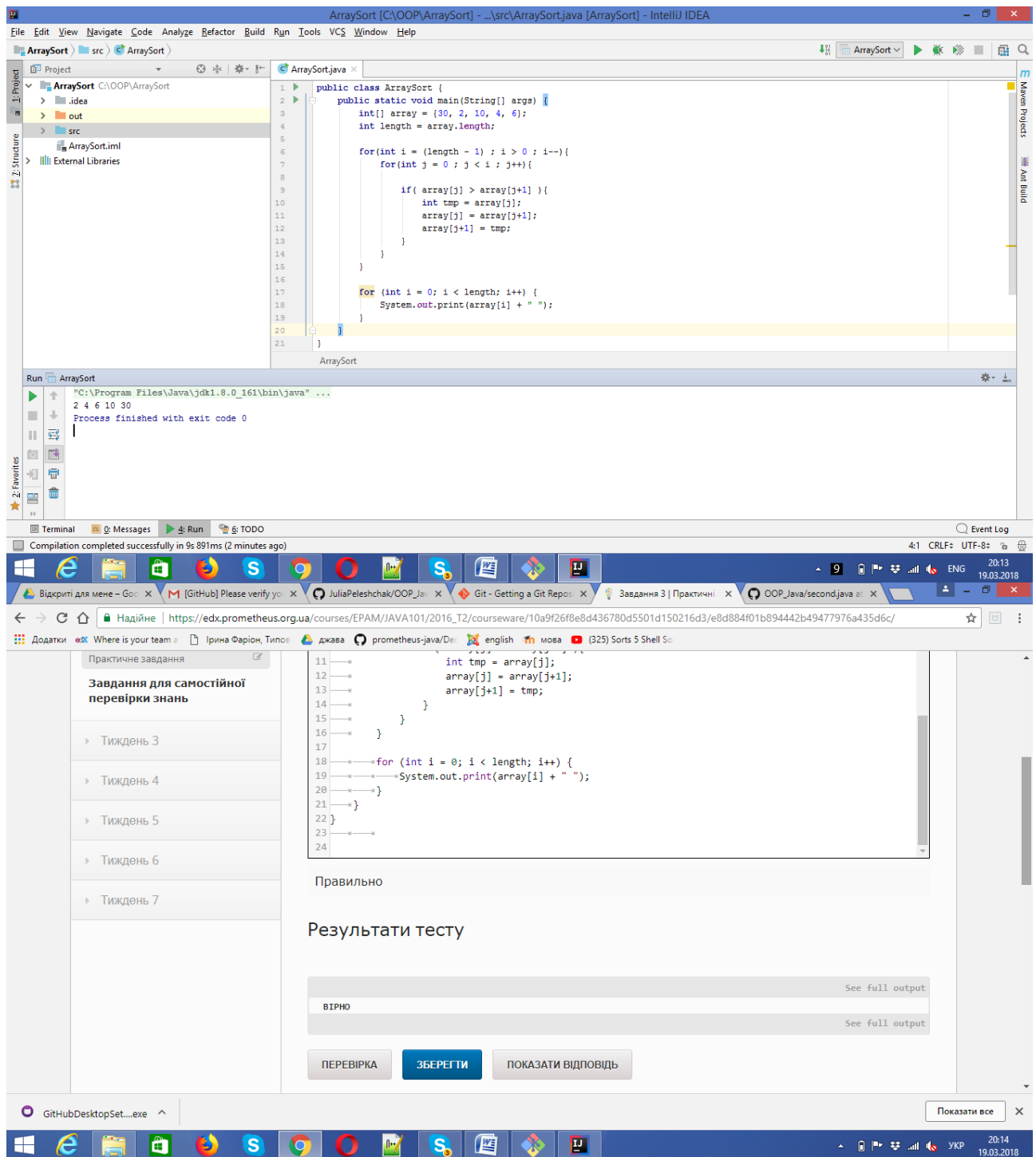




### 3) Напишіть застосування для сортування масиву методом бульбашки

```
public class ArraySort {  
    public static void main(String[] args) {  
        int[] array = {30, 2, 10, 4, 6};  
        //визначаємо довжину масиву  
        int length = array.length;  
  
        for(int i = (length - 1) ; i > 0 ; i--){  
            for(int j = 0 ; j < i ; j++){  
                //якщо попередній елемент більший за наступний, то міняємо їх місцями  
  
                if( array[j] > array[j+1] ){  
                    int tmp = array[j];  
                    array[j] = array[j+1];  
                    array[j+1] = tmp;  
                }  
            }  
        }  
  
        //виводимо посортований масив  
  
        for (int i = 0; i < length; i++) {  
            System.out.print(array[i] + " ");  
        }  
    }  
}
```

Програму зараховано на Прометеусі.



#### 4) Написати застосування для сортування масиву методом сортування Шелла (ShellSort).

```

public class ShellSort {
    public static void main(String[] args) {
        int[] array = {30, 2, 10, 4, 6};

```

```

//визначаємо довжину масиву

```

```

    int length = array.length;

```

```

/* ділимо довжину на 2; порівнюємо елементи між собою, щоразу зменшуючи крок сортування */

```

```

    int a = array.length/2;
    while (a > 0) {
        for (int i = 0; i < (array.length - a); i++) {

```

```

        int s = i;
        while ((s >= 0) && array[s] > array[s + a]) {
            int k = array[s];
            array[s] = array[s + a];
            array[s + a] = k;
            s--;
        }

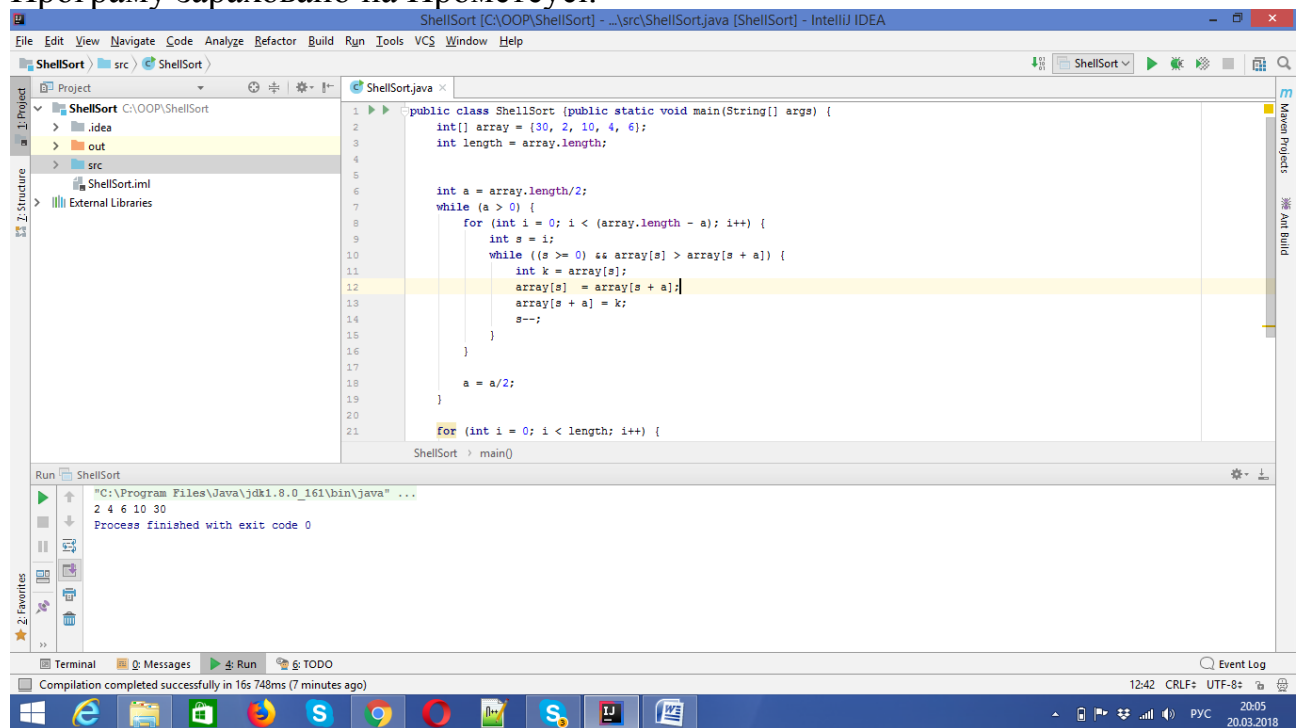
        a = a/2;
    }

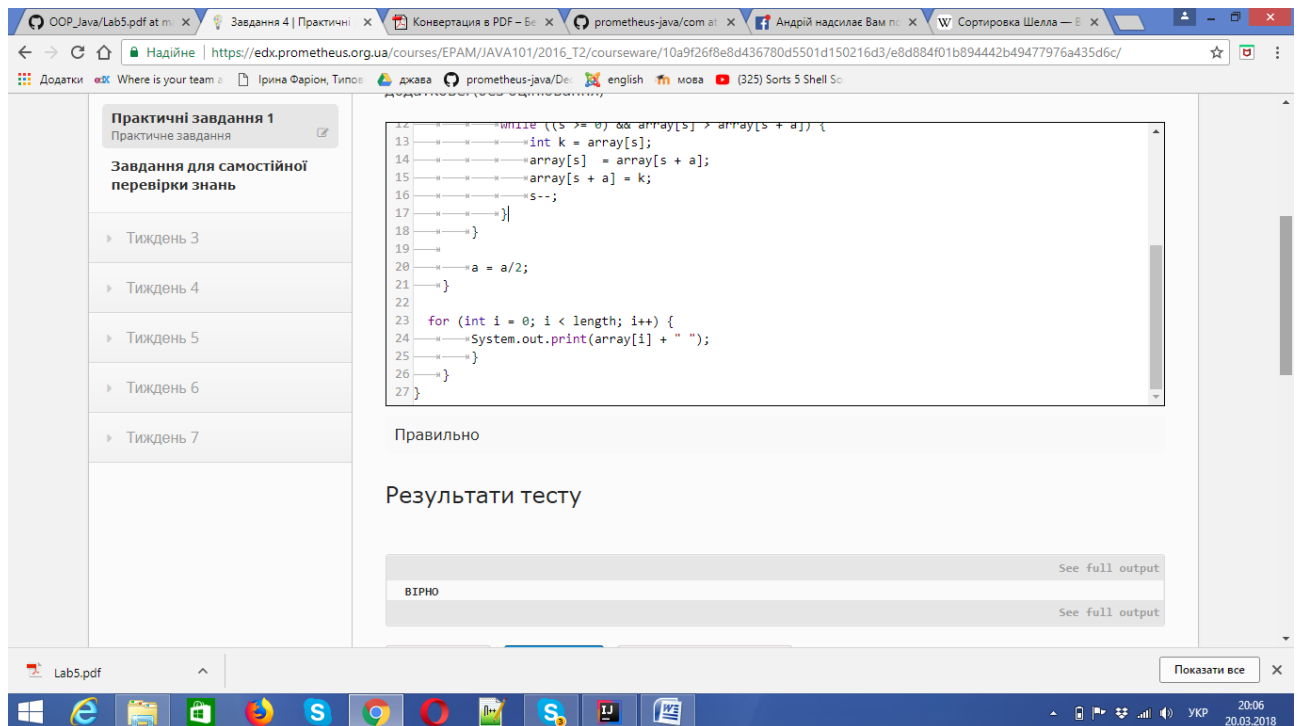
    //Виводимо масив

    for (int i = 0; i < length; i++) {
        System.out.print(array[i] + " ");
    }
}

```

Програму зараховано на Прометеусі.





## 5) Написати застосування, що виконує пошук заданого числа у відсортованому масиві — бінарний пошук.

У випадку коли число знайдено вивести на екран його позицію в масиві (позиції нумеруємо з нуля) або -1 в іншому випадку

```
public class BinarySearch {
    public static void main(String[] args) {

//задаємо масив елементів та число, яке будемо шукати

        int data[] = { 3, 6, 7, 10, 34, 56, 60 };
        int numberToFind = 10;

//визначаємо кінець та початок масиву

        int s = 0;
        int f = data.length - 1;
        int middle = (s + f)/2;

//поки середина не дорівнює 0 виконуємо цикл

        while(s<=f ) {
            middle = (s + f)/2;
//якщо елемент знаходиться в середині, то завершуємо програму

            if (data[middle] == numberToFind ) {
                System.out.println(middle);
                break;
            }

//виконуємо код, якщо елемент знаходиться справа

            if(numberToFind > data[middle]) {
                s = middle + 1;

            }

// виконуємо код, якщо елемент знаходиться зліва
```



```

        if(numberToFind < data[middle]){

            f = middle - 1;

        }

    }

    //якщо не знайшли числа,то виводимо «-1»

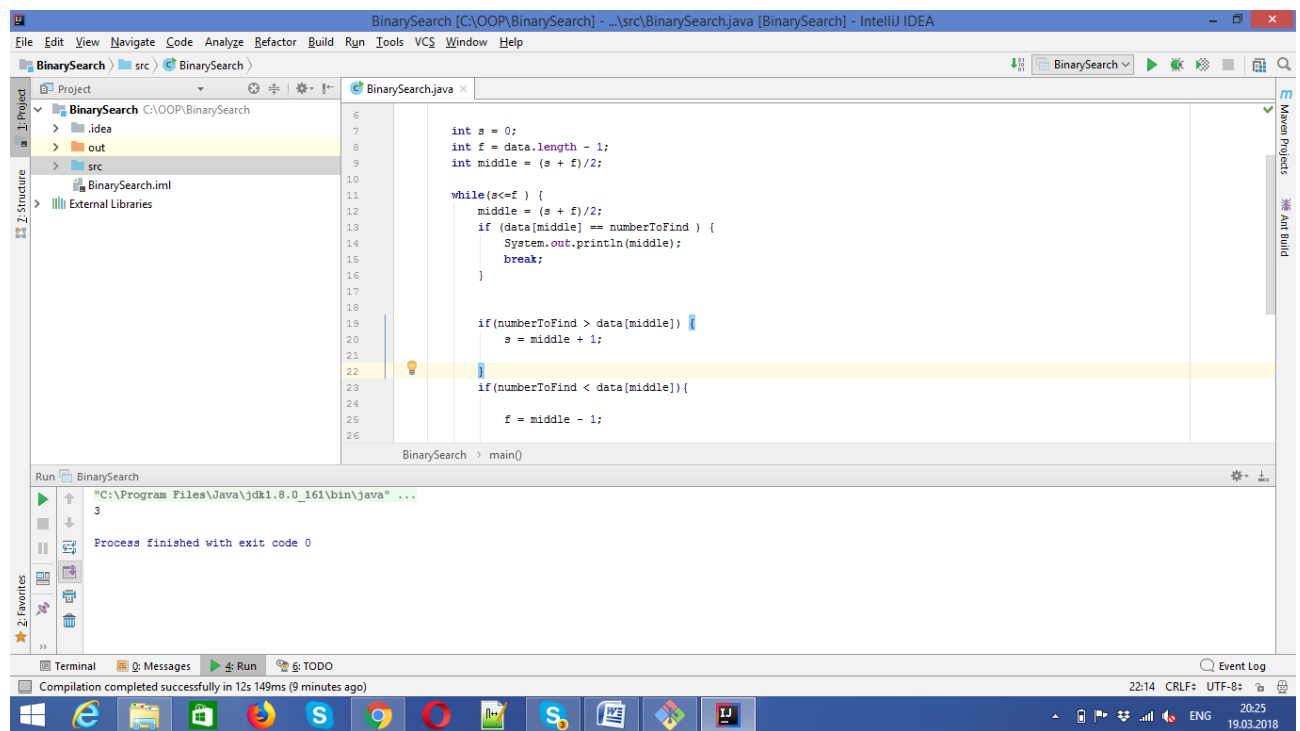
    if(data[middle] != numberToFind) {
        System.out.println(-1);
    }

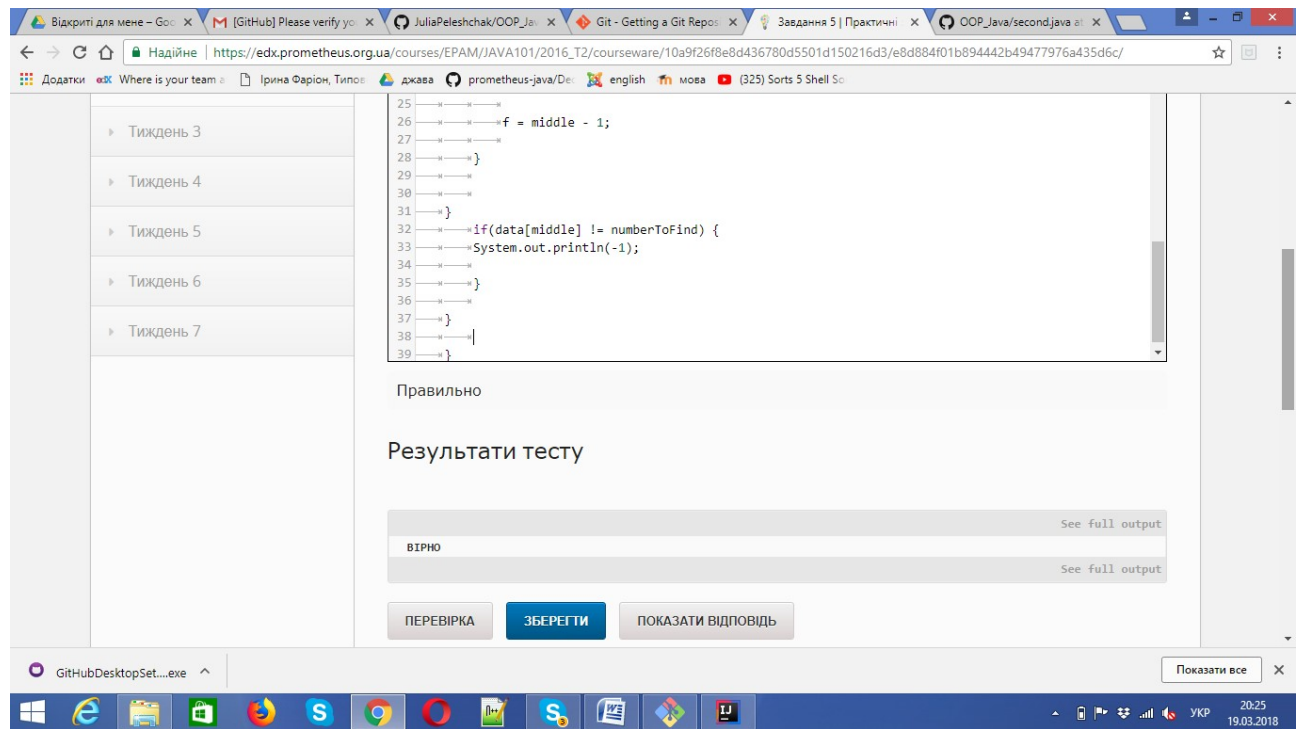
}

}

```

Програму зараховано на Прометеусі.





Висновок: під час виконання лабораторної №2 я засвоїла знання основ мови Java та удосконалила свої знання щодо використання алгоритмів сортування бульбашкою, ShellSort та бінарного пошуку.