

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Лабораторна робота №2
з курсу:
«Об'єктно-орієнтоване програмування»

Виконала:
ст. гр. КН-110
Гелетій Софія
Прийняв:
ст. викладач
Гасько Р.Т.

Львів – 2018 р.

Лабораторна робота № 2

Завдання

Пройти завдання другого тижня на Prometheus на курсі JAVA101.

PRIMETHEUS | автор Ігор Деркач: Основи програмування на Java | sonia_heletiy

Головна сторінка Курс Обговорення Прогрес

Закладки

Вступ до курсу

Тижень 1

Тижень 2

Тема 3. Основи мови

Тест 2

Практичні завдання 1

Завдання для самостійної перевірки знань

Тижень 3

Тижень 4

Тижень 5

Тижень 2 > Тест 2 > Тест 2

Додати до закладок

Перед вами **тест на оцінку** для закріплення знань після **другого тижня навчання**. Оцінка за цей тест буде враховуватися для отримання сертифікату: ви можете отримати до 5 балів за його виконання!

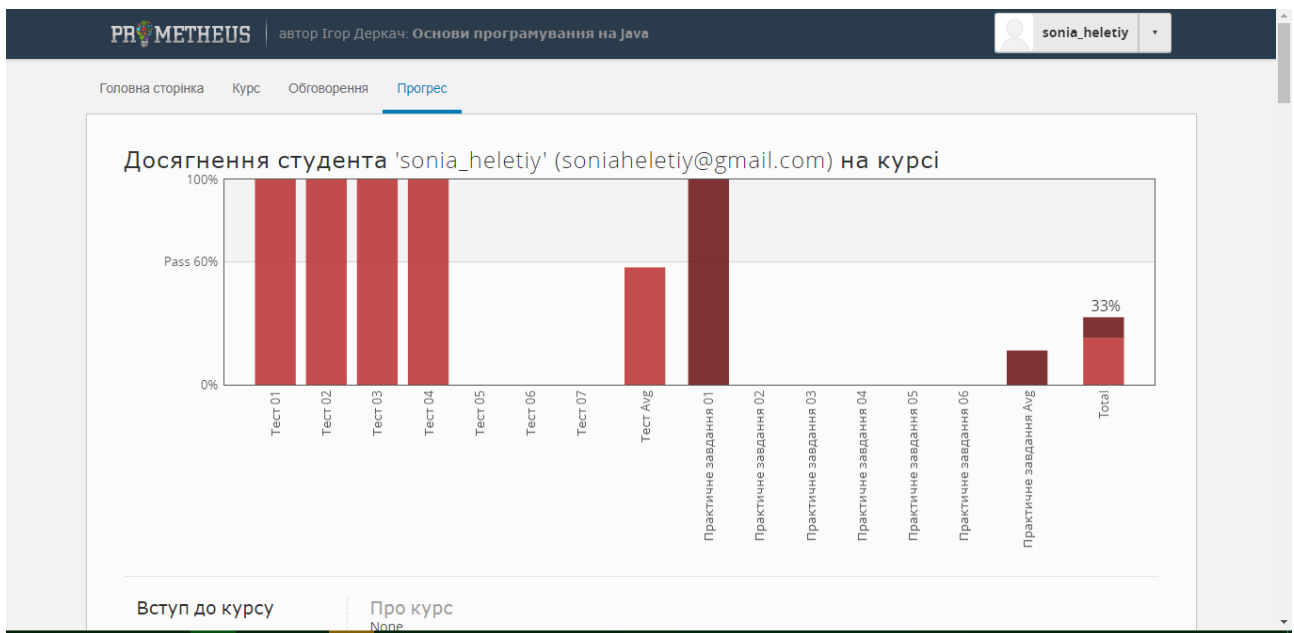
Для виконання завдання у вас є 2 спроби! Скористайтесь кнопкою "Перевірка" для того, щоб надіслати вашу відповідь на перевірку. Кнопка "Зберегти" знадобиться лише в тому випадку, якщо вам потрібно зробити паузу і відійти від комп'ютера. Натисніть на кнопку "Зберегти", щоб зберегти ваші відповіді, але не відправляти їх на перевірку.

Після використання двох спроб - зараховується остання спроба після кнопки "Остаточна відповідь"!

Тест 2

(5/5 балів)

1. Скільки у файлі вихідного коду може бути класів з модифікатором public



1. Створіть застосування що знаходить корені рівняння виду

$$ax^2 + bx + c = 0.$$

```
public class SquareRoot {
```

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

```
        double a = 3;
```

```
        double b = 2.5;
```

```
double c = -0.5;
```

```
double d = b * b - 4 * a * c;
```

```
if (d >= 0) {
```

```
    double x1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
    double x2 = (-b - Math.sqrt(d)) / (2 * a);
```

```
    if (a != 0) {
```

```
        System.out.println("x1=" + x1);
```

```
        System.out.println("x2=" + x2);
```

```
    }
```

```
    if (a == 0 & c == 0 & b != 0) {
```

```
        System.out.println("x1=" + a);
```

```
        System.out.println("x2=" + a);
```

```
    }
```

```
    if (a == 0 & c != 0 & b != 0) {
```

```
        System.out.println("x1=");
```

```
        System.out.println("x2=");
```

```
    }
```

```
    if (a == 0 & b == 0 & c == 0) {
```

```
        System.out.println("x1=");
```

```
        System.out.println("x2=");
```

```
    }
```

```
    if (a == 0 & b == 0 & c != 0) {
```

```
        System.out.println("x1=");
```

```
        System.out.println("x2=");
```

```
    }
```

```
    } else {
```

```
        System.out.println("x1=");
```

```
        System.out.println("x2=");
```

```
    }
```

```
}  
}
```

2. Використовуючи цикл for виведіть на екран матрицю.

```
public class MatrixPrint { public static void main(String args[]){ int ki = 5;  
int kj = 5; String[][] array = new String [][]{
```

```
    {" * ", " 2 ", " 3 ", " 4 ", " * "},  
    {" 6 ", " * ", " 8 ", " * ", "10 "},  
    {"11 ", "12 ", " * ", "14 ", "15 "},  
    {"16 ", " * ", "18 ", " * ", "20 "},  
    {" * ", "22 ", "23 ", "24 ", " * "}};
```

```
for (int i = 0; i < ki; i++) {  
    for (int j = 0; j < kj; j++) {  
        System.out.print(array[i][j]);
```

```
    }  
    System.out.println();  
}  
}
```

3. Напишіть застосування для сортування масиву методом бульбашки

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

```
    int[] array = {30, 2, 10, 4, 6};  
    int length = array.length;
```

```

for (int i = 0; i < length; i++) {
    for (int j = length - 1; j > i; j--) {

        if (array[j] < array[j - 1]) {
            int tmp = array[j];
            array[j] = array[j - 1];
            array[j - 1] = tmp;
        }
    }
}
for (int i = 0; i < length; i++) {
    System.out.print(array[i] + " ");
}
}
}

```

4. Напишіть застосування для сортування масиву методом сортування Шелла (ShellSort).

```

public class ShellSort {

    public static void main(String[] args) {

        int[] array = {30, 2, 10, 4, 6};

        int length = array.length;

        int step = length;
    }
}

```

```

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

{

    boolean ready = true;

    step /= 2;

    for (int k = step; k < length; k++)

    {

        if (array[k] < array[k - step])

        {

            int insert = array[k];

            array[k] = array[k - step];

            int gap = k - step;

            for (int p = gap - step; p >= 0; p -= step)

            {

                if (insert < array[p])

                {

```

```
array[gap] = array[p];
```

```
gap -= step;
```

```
}
```

```
else break;
```

```
}
```

```
array[gap] = insert;
```

```
ready = false;
```

```
}
```

```
}
```

```
if (ready) break;
```

```
}
```

```

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

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

}

}

}

```

5. Напишіть застосування, що виконує пошук заданого числа у відсортованому масиві — бінарний пошук

```

public class BinarySearch {

    public static void main(String[] args) {

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

        int len = data.length;
        boolean flag = false;
        int i;

        for (i = 0; i < len; i++) {
            if (data[i] == numberToFind) {
                flag = true;
                break;
            }
        }

        if (flag == true)
            System.out.println(i);
        else

```



```
        System.out.println("-1");  
    }  
}
```

Висновок: я навчилася писати перші програми на Java, практично закріпила знання на прикладі.