

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту

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

Виконав:

студент групи КН-108

Стасюк Андрій

Викладач:

Гасько Р.Т.

Львів – 2018 р.

Мета:

Ознайомитися з основами мови Java.

Пройти тест 2.

Виконати усі практичні завдання.

Хід роботи

1. Виконав тест правильно.

Тест 2

(5/5 балів)

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

2. Практична частина:

- Завдання 1.

```
public class SquareRoot {
```

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

```
        double a = 3;
```

```
        double b = 2.5;
```

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

```
        double e;
```

```
        e = (b * b) - (4 * a * c);
```

```
        double x = Math.sqrt(e);
```

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

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

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

```
            return;
```

```
        }
```

```
if((a==0) & (b != 0)) {  
    if(c == 0) {  
        double x1 = 0.0;  
        System.out.println("x1="+x1);  
        System.out.println("x2="+x1);  
        return;  
    }  
    else if(c != 0) {  
        double x1 = -(c/b);  
        System.out.println("x1="+x1);  
        System.out.println("x2="+x1);  
        return;  
    }  
    return;  
}
```

```
if(e < 0) {  
    System.out.println("x1="+"\n");  
    System.out.println("x2="+"\n");  
    return;  
}
```

```
if(e >= 0) {  
    double x1 = (-b + x) / (2 * a);  
    double x2 = (-b - x) / (2 * a);  
    System.out.print("x1="+x1+"\n");  
    System.out.println("x2="+x2);  
    return;  
}
```

```

    }

}

}

```

```

14 * * * * *System.out.println("x1=");
15 * * * * *System.out.println("x2=");
16 * * * * *return;
17 * * * * *}
18 * * * * *}
19 * * * * *if((a==0) & (b != 0)) {
20 * * * * *    if(c == 0) {
21 * * * * *        double x1 = 0.0;
22 * * * * *        System.out.println("x1="+x1);
23 * * * * *        System.out.println("x2="+x1);
24 * * * * *        return;
25 * * * * *    }
26 * * * * *    else if(c != 0) {
27 * * * * *        double x1 = -(c/b);
28 * * * * *        System.out.println("x1="+x1);
29 * * * * *        System.out.println("x2="+x1);

```

Правильно

Результати тесту

ВІРНО	See full output
-------	---------------------------------

[ПЕРЕВІРКА](#) [ЗБЕРЕГТИ](#) [ПОКАЗАТИ ВІДПОВІДЬ](#)

- Завдання 2:

```

public class MatrixPrint {
    public static void main(String args[]){
        //PUT YOUR CODE HERE
        int size = 5;
        for(int i = 1; i <= size*size; ++i)
        {
            String printed;
            if( (i==1) | (i==5) | (i==7) | (i==9) | (i==13) | (i==17) | (i==19) | (i==21) |
(i==25))
            {
                printed = " * ";
            }
            else
            {
                printed = String.format("%2s ",i);
            }

```

```

        System.out.print(printed);
        if(i%size == 0)
        {
            System.out.println("");
        }
    }
    //PUT YOUR CODE HERE
}
}

```

```

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

```

```

3 //PUT YOUR CODE HERE
4     int size = 5;
5     for(int i = 1; i <= size*size; ++i)
6     {
7         String printed;
8         if( (i==1) | (i==5) | (i==7) | (i==9) | (i==13) | (i==17) | (i==19) | (i==21) | (i==25))
9         {
10            printed = " * ";
11        }
12        else
13        {
14            printed = String.format("%2s ",i);
15        }
16
17        System.out.print(printed);
18        if(i%size == 0)

```

Правильно

Результати тесту

ВІРНО

[See full output](#)

[See full output](#)

- Завдання 3

```

public class ArraySort {

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

        //PUT YOUR CODE HERE
        int n = length;
        int tmp;
        for(int j = 0; j < n - 1; j++){
            for(int i = 0; i < n - 1; i++){
                if (array[i] > array[i + 1])
                {
                    tmp = array[i];

```

```

        array[i] = array[i + 1];
        array[i + 1] = tmp;
    }
}
}

//PUT YOUR CODE HERE

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

```

Завдання 3

(2/2 бали)

Напишіть застосування для сортування масиву методом бульбашки

```

1 public class ArraySort {
2
3     public static void main(String[] args) {
4         int[] array = {30, 2, 10, 4, 6};
5         int length = array.length;
6
7         //PUT YOUR CODE HERE
8         int n = length;
9         int tmp;
10        for(int j = 0; j < n - 1; j++){
11            for(int i = 0; i < n - 1; i++){
12                if (array[i] > array[i + 1])
13                {
14                    tmp = array[i];
15                    array[i] = array[i + 1];
16                    array[i + 1] = tmp;
17                }
18            }
19        }
20    }
21 }

```

Правильно

Результати тесту

ВІРНО	See full output
	See full output

- **Завдання 5:**

```

public class BinarySearch {

    public static void main(String[] args) {

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

        // PUT YOUR CODE HERE

        int low = 0;
        int high = data.length - 1;
    }
}

```

```

while (high >= low) {
    int middle = (low + high) / 2;

    if (data[middle] == numberToFind)
    {
        System.out.println(middle);
        return;
    }
    else if (data[middle] < numberToFind)
    {
        low = middle + 1;
    }
    else if (data[middle] > numberToFind)
    {
        high = middle - 1;
    }
}
System.out.println(-1);
return;

        // PUT YOUR CODE HERE
    }
}

```

Завдання 5

(2/2 бали)

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

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

```

1 public class BinarySearch {
2
3     public static void main(String[] args) {
4
5         int data[] = { 3, 6, 7, 10, 34, 56, 60 };
6         int numberToFind = 10;
7
8         // PUT YOUR CODE HERE
9
10        int low = 0;
11        int high = data.length - 1;
12
13        while (high >= low) {
14            int middle = (low + high) / 2;
15
16            if (data[middle] == numberToFind)

```

Правильно

Результати тесту

3. Також було додаткове самостійне завдання пов'язане з реалізацією бінарного дерева:

package BinaryTreeOOP;

```
public class BinaryTree {  
    BinaryTree tree;  
    Node root;  
  
    public BinaryTree create(int[] arr){  
        tree = new BinaryTree();  
        for (int i = 0; i < arr.length; i++){  
            tree.addNode(arr[i]);  
        }  
        return tree;  
    }  
  
    private Node add(int value, Node root){  
        if (root == null){  
            return new Node(value);  
        }  
  
        if (value < root.getValue()){  
            root.setLeft(add(value, root.getLeft()));  
        }  
  
        if (value > root.getValue()){  
            root.setRight(add(value, root.getRight()));  
        }  
  
        return root;  
    }  
}
```



```
public void addNode(int value) {  
    root = add(value, root);  
}
```

```
private boolean search(int value, Node root){  
    if (root == null){  
        return false;  
    }
```

```
    if (value < root.getValue()){  
        return search(value, root.getLeft());  
    }
```

```
    if (value > root.getValue()){  
        return search(value, root.getRight());  
    }
```

```
    if (value == root.getValue()){  
        return true;  
    }
```

```
    return false;  
}
```

```
public boolean searchNode(int value){  
    return search(value, root);  
}  
}
```

```
package BinaryTreeOOP;
```

```
public class Main {
```

```
    public static void main(String[] args) {  
        BinaryTree tree = new BinaryTree();  
        int[] arr = {3,18,2,6,100,19,27,1,17};  
        tree = tree.create(arr);  
        System.out.println(tree.searchNode(17));  
        System.out.println(tree.searchNode(111));  
        System.out.println(tree.searchNode(134));  
        tree.addNode(1);  
        System.out.println(tree.searchNode(17));  
        System.out.println(tree.searchNode(111));  
        System.out.println(tree.searchNode(134));  
    }  
}
```

```
package BinaryTreeOOP;
```

```
public class Node {  
    private int value;  
    private Node left;  
    private Node right;  
  
    public Node(int value)  
    {  
        setValue(value);  
        right = null;  
        left = null;  
    }  
}
```

```
}  
public int getValue()  
{  
    return value;  
}  
public void setValue(int value)  
{  
    this.value = value;  
}  
public Node getLeft()  
{  
    return left;  
}  
public void setLeft(Node left)  
{  
    this.left = left;  
}  
public Node getRight()  
{  
    return right;  
}  
public void setRight(Node right)  
{  
    this.right = right;  
}  
}
```

Прогрес студента

