

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 7

з дисципліни: «Кросплатформні засоби програмування»

«Параметризоване програмування»

Варіант - 1

Виконав:  
ст. гр. КІ-34  
Віщур Н.О  
Прийняв:  
Олексів М. В.

Львів 2022

## ЗАВДАННЯ

1. Створити параметризований клас, що реалізує предметну область задану варіантом. Клас має містити мінімум 4 методи опрацювання даних включаючи розміщення та виймання елементів. Парні варіанти реалізують пошук мінімального елементу, непарні – максимального. Написати на мові Java та налагодити програму-драйвер для розробленого класу, яка мстить мінімум 2 різні класи екземпляри яких розмішуються у екземплярі розробленого класу-контейнеру. Програма має розміщуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

### Варіант завдання:

## 1. Масив

### Код програми:

#### File App.java

```
package KI34.Vishchur.lab7;

public class App {
    /**
     * @param args
     * @throws Exception
     */
    public static void main(String[] args) throws Exception
    {
        // Ініціалізація масиву
        Array<? super Shape> mall = new Array();
        // Додаємо фігури
        mall.add(new Sphere(1000, 100,100, "Sphere1"));
        mall.add(new Cube(1000, 10,10, "Cube1" ));
        // Виводимо
        mall.showAll();
        // Виводимо першу
        System.out.println(mall.get(0).getName());
        // Сукупний об'єм
        System.out.println(mall.getTotalVolume());
        // Найбільша фігура
        System.out.println(mall.getLargestFigure().getName());
    }
}
```

```
}
```

## File Array.java

```
package KI34.Vishchur.lab7;

import java.util.ArrayList;
/**
 * Class Array
 * @version 1.0
 */
public class Array<T extends Shape> {
    private ArrayList<T> storage = new ArrayList();
    /**
     * Method returns an item by the index
     * @param index Element index
     * @throws Exception
     */
    public T get(int index) throws Exception
    {
        try{
            return storage.get(index);
        }
        catch (Exception e)
        {
            throw e;
        }
    }
    /**
     * Method adds an item
     * @param item Item to be added
     * @throws Exception
     */
    public void add(T item) throws Exception
    {
        try {
            storage.add(item);
        }
        catch (Exception e)
        {
            throw e;
        }
    }
    /**
     * Method shows storage content
     */
    public void showAll()
```

```

    {
        for(int i = 0; i < storage.size(); i++)
        {
            try{
                storage.get(i).printData();
            }
            catch (Exception e)
            {
                System.out.println(e);
            }
        }
    }

    /**
     * Method returns total volume
     */
    public int getTotalVolume()
    {
        int result = 0;
        for(int i = 0; i < storage.size(); i++)
        {
            result += storage.get(i).getVolume();
        }
        return result;
    }

    /**
     * Method returns the biggest item
     */
    public T getLargestFigure()
    {
        int largestIdx = 0;

        for(int i = 0; i < storage.size(); i++)
        {
            if(i == 0)
            {
                continue;
            }
            if(storage.get(i).getVolume() >
storage.get(largestIdx).getVolume())
            {
                largestIdx = i;
            }
        }
        return storage.get(largestIdx);
    }
}

```

## File Shape.java

```
package KI34.Vishchur.lab7;

public class Shape {
    private int volume;
    private int width;
    private int height;
    private String name;
    /**
     * Constructor
     * @param volume
     * @param width
     * @param height
     * @param name
     */
    public Shape(int volume, int width, int height, String
name)
    {
        this.volume = volume;
        this.width = width;
        this.height = height;
        this.name = name;
    }
    /**
     * Method prints shape data
     */
    public void printData()
    {
        System.out.println("Name: " + name);
        System.out.println("Volume: " + volume);
        System.out.println("Width: " + width);
        System.out.println("Height: " + height);
    }
    /**
     * Method returns shape height
     */
    public int getHeight(){ return height; }
    /**
     * Method returns shape volume
     */
    public int getVolume(){ return volume; }
    /**
     * Method returns shape width
     */
    public int getWidth() { return width; }
```

```

    /**
     * Method returns shape name
     */
    public String getName() { return name; }
}

```

### File Cube.java

```

package KI34.Vishchur.lab7;

public class Cube extends Shape {
    /**
     * Constructor
     * @param volume
     * @param width
     * @param height
     * @param name
     */
    public Cube(int volume, int width, int height, String
name)
    {
        super(volume, width, height, name);
    }
}

```

### File Sphere.java

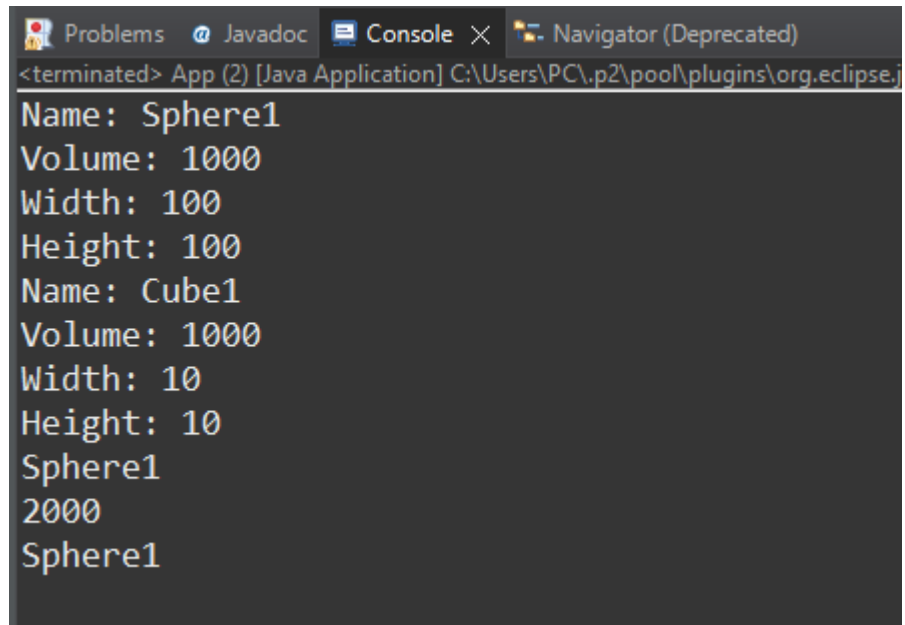
```

package KI34.Vishchur.lab7;

public class Sphere extends Shape {
    /**
     * Constructor
     * @param volume
     * @param width
     * @param height
     * @param name
     */
    public Sphere(int volume, int width, int height, String
name)
    {
        super(volume, width, height, name);
    }
}

```

**Результат виконання роботи:**



```
Problems Javadoc Console X Navigator (Deprecated)
<terminated> App (2) [Java Application] C:\Users\PC\.p2\pool\plugins\org.eclipse.j
Name: Sphere1
Volume: 1000
Width: 100
Height: 100
Name: Cube1
Volume: 1000
Width: 10
Height: 10
Sphere1
2000
Sphere1
```

**Висновок:** під час виконання цієї роботи я оволодів навичками параметризованого програмування мовою Java.