

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних технологій, автоматики та метрології

Кафедра ЕОМ



Звіт
До лабораторної роботи №7
З дисципліни: «Кросплатформні засоби програмування»
На тему «Параметризоване програмування»
Варіант №9

Виконав: ст. гр. КІ-36
Нагребний Ю.С.

Прийняв:
Іванов Ю.С.

Львів – 2022

Мета: оволодіти навиками параметризованого програмування мовою Java.

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

8

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.

9. Споруда

Виконання:

Лістинг програми:

```
//Main.java
package KI36.Nahrebnyi.Lab7;

public class Main
{
    public static void main(String[] args)
    {
        Building<? super Room> building = new Building<>();

        building.AddItem(new Flat(45.4, 4, 3));
        building.AddItem(new Flat(41.2, 7, 2));
        building.AddItem(new Flat(22.6, 6, 3));
        building.AddItem(new Flat(85.2, 2, 5));
        building.AddItem(new Duplex(45.4, 4, 334567, 3));
        building.AddItem(new Duplex(127.4, 9, 100000, 5));
        building.AddItem(new Duplex(109.2, 10, 902000, 4));
        building.AddItem(new Duplex(134.9, 15, 33456227, 6));
        building.AddItem(new Duplex(121.4, 10, 9630000, 4));
    }
}
```

```

        var max = building.FindMax();
        System.out.println("All buildings: ");
        building.PrintAllItems();

        System.out.print("Max element: ");
        max.PrintInfo();
    }
}

//Room.java
package KI36.Nahrebnyi.Lab7;

public interface Room
{
    double getArea();
    void PrintInfo();
}

//Flat.java
package KI36.Nahrebnyi.Lab7;

public class Flat implements Room
{
    private double area;
    private int floor;
    private int numberOfRooms;

    public Flat(double area, int floor, int numberOfRooms)
    {
        this.area = area;
        this.floor = floor;
        this.numberOfRooms = numberOfRooms;
    }

    @Override
    public double getArea()
    {
        return area;
    }

    @Override
    public void PrintInfo()
    {
        System.out.println("Flat: -area: " + area + "; -floor: " + floor + "; -
number of rooms: " + numberOfRooms);
    }

    public void setArea(double area)
    {
        this.area = area;
    }

    public int getFloor()
    {
        return floor;
    }
}

```

```

    }

    public void setFloor(int floor)
    {
        this.floor = floor;
    }

    public int getNumberOfRooms()
    {
        return numberOfRooms;
    }

    public void setNumberOfRooms(int numberOfRooms) {
        this.numberOfRooms = numberOfRooms;
    }
}

//Duplex.java
package KI36.Nahrebnyi.Lab7;

public class Duplex implements Room
{
    private double area;
    private int floor;
    private int numberOfRooms;
    private double price;

    public Duplex(double area, int floor, double price, int numberOfRooms)
    {
        this.area = area;
        this.floor = floor;
        this.price = price;
        this.numberOfRooms = numberOfRooms;
    }

    public void setArea(double area)
    {
        this.area = area;
    }

    public int getFloor()
    {
        return floor;
    }

    public void setFloor(int floor)
    {
        this.floor = floor;
    }

    public double getPrice()
    {
        return price;
    }

    public void setPrice(double price)
    {
        this.price = price;
    }
}

```

```

    }

    public int getNumberOfRooms()
    {
        return numberOfRooms;
    }

    public void setNumberOfRooms(int numberOfRooms)
    {
        this.numberOfRooms = numberOfRooms;
    }

    @Override
    public double getArea()
    {
        return area;
    }

    @Override
    public void PrintInfo()
    {
        System.out.println("Duplex: -area: " + area + "; -floor: " + floor + ";
        -number of rooms: " + numberOfRooms + "; -price: " + price);
    }
}

//Building.java
package KI36.Nahrebnyi.Lab7;

import java.util.ArrayList;
import java.util.List;

public class Building <T extends Room>
{
    private final List<T> buildings;

    public Building()
    {
        this.buildings = new ArrayList<T>();
    }

    public T FindMax()
    {
        if (buildings.isEmpty())
        {
            return null;
        }
        else
        {
            T max = buildings.get(0);
            for (int i = 1; i < buildings.size(); i++)
            {
                if (buildings.get(i).getArea() > max.getArea())
                {
                    max = buildings.get(i);
                }
            }
            return max;
        }
    }
}

```

```

    }

    public void AddItem(T item)
    {
        buildings.add(item);
        var index = buildings.indexOf(item);
        System.out.print("Element was added: index - " + index + "; ");
        item.PrintInfo();
    }

    public void DeleteItem(int i)
    {
        System.out.print("Element on index " + i + " was deleted: ");
        buildings.get(i).PrintInfo();
        buildings.remove(i);
    }

    public void PrintAllItems()
    {
        for (T el : buildings)
        {
            el.PrintInfo();
        }
    }
}

```

Результати:

```

All buildings:
Flat: -area: 45.4; -floor: 4; -number of rooms: 3
Flat: -area: 41.2; -floor: 7; -number of rooms: 2
Flat: -area: 22.6; -floor: 6; -number of rooms: 3
Flat: -area: 85.2; -floor: 2; -number of rooms: 5
Duplex: -area: 45.4; -floor: 4; -number of rooms: 3; -price: 334567.0
Duplex: -area: 127.4; -floor: 9; -number of rooms: 5; -price: 100000.0
Duplex: -area: 109.2; -floor: 10; -number of rooms: 4; -price: 902000.0
Duplex: -area: 134.9; -floor: 15; -number of rooms: 6; -price: 3.3456227E7
Duplex: -area: 121.4; -floor: 10; -number of rooms: 4; -price: 9630000.0
Max element: Duplex: -area: 134.9; -floor: 15; -number of rooms: 6; -price: 3.3456227E7

Process finished with exit code 0

```

Висновок: оволодів навиками параметризованого програмування мовою Java.