

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 №16

Виконав: ст. гр. КІ-36
Струтинський В.О.
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 розміщуються 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.

16. Земельна ділянка

Виконання:

Лістинг програми:

```
public interface Building
{
    double GetSquare();
    void PrintInfo();
}
public class House implements Building
{
    private double square;
    private String address;

    public House(double square, String address) {
        this.square = square;
        this.address = address;
    }

    public double getSquare() {
        return square;
    }

    public void setSquare(double square) {
        this.square = square;
    }
}
```

```

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public double GetSquare() {
        return square;
    }

    @Override
    public void PrintInfo() {
        System.out.println("House { square = " + square + "; address = " +
address + "; }");
    }
}
public class Factory implements Building
{
    private double square;
    private String address;
    private String companyName;

    public Factory(double square, String address, String companyName)
    {
        this.square = square;
        this.address = address;
        this.companyName = companyName;
    }

    public double getSquare() {
        return square;
    }

    public void setSquare(double square) {
        this.square = square;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getCompanyName() {
        return companyName;
    }

    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    @Override
    public double GetSquare() {
        return square;
    }

    @Override

```

```

        public void PrintInfo() {
            System.out.println("Factory { square = " + square + "; address = " +
address + "; company name " + companyName + ";}");
        }
    }
import java.util.ArrayList;
import java.util.List;

public class LandPlot<T extends Building>
{
    private final List<T> arr;

    public LandPlot()
    {
        arr = new ArrayList<T>();
    }

    public T FindMin()
    {
        if (arr.isEmpty())
        {
            return null;
        }
        else
        {
            T min = arr.get(0);
            for (int i = 1; i < arr.size(); i++)
            {
                if (arr.get(i).GetSquare() < min.GetSquare())
                {
                    min = arr.get(i);
                }
            }
            return min;
        }
    }

    public void AddItem(T item)
    {
        arr.add(item);
        var index = arr.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: ");
        arr.get(i).PrintInfo();
        arr.remove(i);
    }

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

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

```

```

{
    LandPlot<? super Building> landPlot = new LandPlot<>();

    landPlot.AddItem(new House(56.7, "Dfdfdfff"));
    landPlot.AddItem(new House(12, "Dfdfdfff"));
    landPlot.AddItem(new Factory(521, "Dfdfdfff", "fefe"));
    landPlot.AddItem(new Factory(23, "Dfdfdfff", "efefe"));
    landPlot.AddItem(new House(433, "Dfdfdfff"));
    landPlot.AddItem(new Factory(43, "Dfdfdfff", "efgtrgr"));
    landPlot.AddItem(new House(5435, "Dfdfdfff"));
    landPlot.AddItem(new Factory(511, "Dfdfdfff", "grgrgr"));

    System.out.println("Min element is: " );
    landPlot.FindMin().PrintInfo();
}
}

```

Результати:

```

Element was added: index - 0; House { square - 56.7; address - Dfdfdfff; }
Element was added: index - 1; House { square - 12.0; address - Dfdfdfff; }
Element was added: index - 2; Factory { square - 521.0; address - Dfdfdfff; company name fefe;}
Element was added: index - 3; Factory { square - 23.0; address - Dfdfdfff; company name efefe;}
Element was added: index - 4; House { square - 433.0; address - Dfdfdfff; }
Element was added: index - 5; Factory { square - 43.0; address - Dfdfdfff; company name efgtrgr;}
Element was added: index - 6; House { square - 5435.0; address - Dfdfdfff; }
Element was added: index - 7; Factory { square - 511.0; address - Dfdfdfff; company name grgrgr;}
Min element is:
House { square - 12.0; address - Dfdfdfff; }

```

Висновок: оволодів навиками параметризованого програмування мовою Java.