

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №3
з дисципліни «Інженерія програмного забезпечення»
на тему: «Структурні шаблони проектування.
Шаблони Composite, Decorator, Proxy»

Виконав:
студент 2-го курсу ФІОТ
групи ІВ-71
Мазан Я.В.
Номер залікової книжки: 7109
Варіант: 5

Перевірив:
доцент кафедри ОТ
Антонюк А.І.

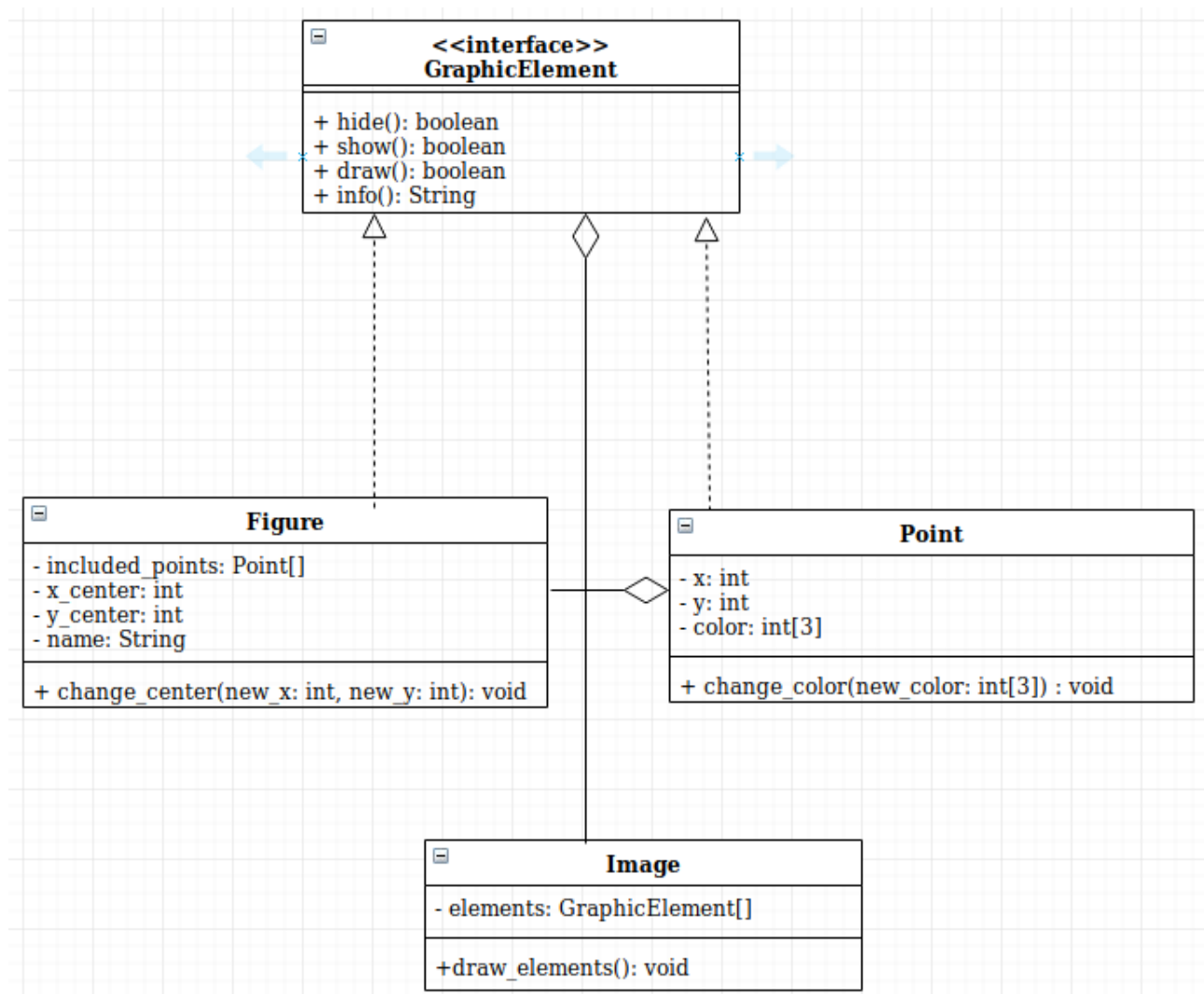
Київ 2018

Завдання

Варіант – 5

Визначити специфікації класів та реалізацію методів для подання вибраного графічного елементу у редакторі векторної графіки. Забезпечити можливість динамічної зміни відображення елементу.

Діаграма класів



Код програми

Main.java:

```
package com.lab111.labwork3;
/**
 * Lab 3 of ingeneering of programming software
 * @author Yan Mazan
 */
public class Main {
    /**
     * Main method
     * @param args - arguments of the program if is executed in terminal
     */
    public static void main(String[] args) {
```

```

        Point random_point = new Point(2,3,new int[]{13,24,155});
        Point[] figure_points = new Point[]{new Point(1,1,new int[]{3,3,12}),new Point(0,0,new
int[]{0,213,125})};
        Figure random_figure = new Figure(2,5,figure_points, "Two points");
        Image img = new Image(new GraphicElement[]{random_point,random_figure});
        img.draw();
    }
}

```

GraphicElement.java:

```

package com.lab111.labwork3;
/**
 * Interface of basic graphic element of the image
 * Is implemented by Point and Figure classes
 */
public interface GraphicElement {
    boolean hide();
    boolean show();
    boolean draw();
    String info();
}

```

Image.java:

```

package com.lab111.labwork3;
public class Image {
    private GraphicElement[] elements;
    public Image(GraphicElement[] elements_of_image) {
        elements = elements_of_image;
    }
    /**
     * Method to draw every element of the image
     */
    public void draw() {
        for (GraphicElement i : elements) {
            i.draw();
        }
    }
}

```

Point.java:

```

package com.lab111.labwork3;
public class Point implements GraphicElement{
    private int x;
    private int y;
    private int[] color;
    public Point(int x, int y, int[] color) {
        this.color = new int[3];
        this.color = color;
        this.x = x;
        this.y = y;
    }
    /**
     * Method to hide a point
     * Realises hide() method of implemented interface
     * @return - always true because the element can always be hidden
     */
    public boolean hide() {
        System.out.println(info() + " was hidden;");
        return true;
    }
}
/**

```

```

    * Method to show a point
    * Realises show() method of implemented interface
    * @return - always true because the element can always be shown
    */
    public boolean show() {
        System.out.println(info() + " was shown;");
        return true;
    }
    /**
    * Method to draw a point
    * Realises draw() method of implemented interface
    * @return - always true because the element can always be drawn
    */
    public boolean draw() {
        System.out.println(info() + " was drawn;");
        return true;
    }
    /**
    * Method to return information about a point
    * Realises info() method of implemented interface
    * @return - point coordinates and its' color
    */
    public String info() {
        return "Point with x = " + x + " y = " + y + " with color [" + color[0] + ", " +
color[1] + ", " + color[2] + "];"
    }
    /**
    * Method that changes a color of a point to a new one
    * @param new_color - new color to assign to a point
    */
    public void change_color(int[] new_color) {
        this.color = new int[3];
        this.color = new_color;
    }
}

```

Figure.java:

```

package com.lab111.labwork3;
public class Figure implements GraphicElement{
    private Point[] included_points;
    private int x_center;
    private int y_center;
    private String name;
    public Figure(int x, int y, Point[] points, String figure_name) {
        included_points = points;
        x_center = x;
        y_center = y;
        name = figure_name;
    }
    /**
    * Method to return information about a figure
    * Realises info() method of implemented interface
    * @return - figure coordinates and its' name
    */
    public String info() {
        return "Figure with name " + name + " x center = " + x_center + " y center = " +
y_center;
    }
    /**
    * Method to draw a figure
    * Realises hide() method of implemented interface
    * @return - always true because the element can always be hidden
    */
    public boolean hide() {

```

```

        System.out.println(info() + " was hidden;");
        return true;
    }
    /**
     * Method to show a figure
     * Realises show() method of implemented interface
     * @return - always true because the element can always be shown
     */
    public boolean show() {
        System.out.println(info() + " was shown;");
        return true;
    }
    /**
     * Method to draw a figure
     * Realises draw() method of implemented interface
     * @return - always true because the element can always be drawn
     */
    public boolean draw() {
        System.out.println(info() + " was drawn\nPoints inside the figure " + name +
        ":");
        for (Point i: included_points) {
            i.draw();
        }
        System.out.println("were also drawn;");
        return true;
    }
    /**
     * Changes the center of a figure to a new one
     * @param new_x - new x center
     * @param new_y - new y center
     */
    public void change_center(int new_x, int new_y) {
        x_center = new_x;
        y_center = new_y;
    }
}

```

Висновок

При виконанні завдання лабораторної роботи було використано шаблон Composite, суть використання якого полягає у використанні спільного інтерфейсу для базового та композитного класу для того, щоб однаково їх обробляти під час виконання програми. Також під час вибору шаблону для реалізації мого завдання довелось ознайомитись та вивчити шаблони Decorator та Proxy. Програма виконана успішно та реалізовує поставлену задачу.