

Харківський національний університет імені В.Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра електроніки та управляючих систем

Звіт  
з контрольної роботи  
дисципліна: «Проектування інформаційних систем»

Виконав: студент групи КС34  
Мельник Д.А.  
Перевірів: доцент  
Гамзаєв Р. О.

Харків 2022

## **Варіант №2**

Структурний паттерн Flyweight представляє з себе паттерн проектування, який дозволяє вмістити більшу кількість об'єктів в виділену оперативну пам'ять. Флайвейт економить пам'ять, розділяючи загальний стан об'єктів між собою, замість збереження однакових даних в кожному об'єкті.

Тобто, якщо якийсь шутер з системою частинок на доволі потужному ПК буде працювати добре, то на слабкому ПК він може одразу вилетіти, саме через нестачу пам'яті. Для цього й існує Флайвейт – аби зберігати частинки з однаковими параметрами разом, а не кожен окремо.

В моїй програмі показаний саме цей патерн, на прикладі точок різних кольорів, кількість яких можна змінити та налаштувати кольори.

## Лістинг 1: Demo.java

```
package Flyweight;

import Flyweight.Field.Field;

import java.awt.*;

public class Demo {
    static int CANVAS_SIZE = 1000;
    static int DOTS_TO_DRAW = 1000000;
    static int DOTS_TYPES = 11;

    public static void main(String[] args) {
        Field field = new Field();
        for (int i = 0; i < Math.floor(DOTS_TO_DRAW / DOTS_TYPES); i++) {
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Green", Color.GREEN, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Orange", Color.ORANGE, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Blue", Color.BLUE, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Red", Color.RED, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Pink", Color.PINK, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Yellow", Color.YELLOW, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Cyan", Color.CYAN, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Black", Color.BLACK, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Magenta", Color.MAGENTA, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Light Gray", Color.LIGHT_GRAY, "");
            field.plantDot(random(0, CANVAS_SIZE), random(0, CANVAS_SIZE),
                "Dark Gray", Color.DARK_GRAY, "");
        }
        field.setSize(CANVAS_SIZE, CANVAS_SIZE);
        field.setVisible(true);

        System.out.println(DOTS_TO_DRAW + " dots drawn");
        System.out.println("-----");
        System.out.println("Memory usage:");
        System.out.println("Dot size (8 bytes) * " + DOTS_TO_DRAW);
        System.out.println("+ DotTypes size (~30 bytes) * " + DOTS_TYPES +
            "");
        System.out.println("-----");
        System.out.println("Total: " + ((DOTS_TO_DRAW * 8 + DOTS_TYPES * 30)
            / 1024 / 1024) +
            "MB (instead of " + ((DOTS_TO_DRAW * 38) / 1024 / 1024) +
            "MB)");
    }

    private static int random(int min, int max) {
        return min + (int) (Math.random() * ((max - min) + 1));
    }
}
```

## Лістинг 2: Dot.java

```
package Flyweight.Dots;

import java.awt.*;

public class Dot {
    private int x;
    private int y;
    private DotType type;

    public Dot(int x, int y, DotType type) {
        this.x = x;
        this.y = y;
        this.type = type;
    }

    public void draw(Graphics g) {
        type.draw(g, x, y);
    }
}
```

## Лістинг 3: DotFactory.java

```
package Flyweight.Dots;

import java.awt.*;
import java.util.HashMap;
import java.util.Map;

public class DotFactory {
    static Map<String, DotType> dotTypes = new HashMap<>();

    public static DotType getDotType(String name, Color color, String
otherDotData) {
        DotType result = dotTypes.get(name);
        if (result == null) {
            result = new DotType(name, color, otherDotData);
            dotTypes.put(name, result);
        }
        return result;
    }
}
```

## Лістинг 4: DotType.java

```
package Flyweight.Dots;

import java.awt.*;

public class DotType {
    private String name;
    private Color color;
    private String otherDotData;

    public DotType(String name, Color color, String otherDotData) {
        this.name = name;
        this.color = color;
        this.otherDotData = otherDotData;
    }

    public void draw(Graphics g, int x, int y) {
        g.setColor(color);
        g.fillOval(x - 5, y - 10, 10, 10);
    }
}
```

## Лістинг 5: Field.java

```
package Flyweight.Field;

import Flyweight.Dots.Dot;
import Flyweight.Dots.DotFactory;
import Flyweight.Dots.DotType;

import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
import java.util.List;

public class Field extends JFrame {
    private List<Dot> dots = new ArrayList<>();

    public void plantDot(int x, int y, String name, Color color, String
otherDotData) {
        DotType type = DotFactory.getDotType(name, color, otherDotData);
        Dot dot = new Dot(x, y, type);
        dots.add(dot);
    }

    @Override
    public void paint(Graphics graphics) {
        for (Dot dot : dots) {
            dot.draw(graphics);
        }
    }
}
```

## Діаграма класів:

