



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

Лабораторна робота №8

із дисципліни «Технології розроблення програмного забезпечення»

Тема: «ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»»

Перевірив:
Мягкий М.Ю

Виконала:
Студентка групи ІА-24
Ганжа Х.М

Варіант:

..4 Графічний редактор (proxy, prototype, decorator, bridge, flyweight, SOA)
Графічний редактор повинен вміти створювати / редагувати растрові (або векторні на розсуд студента) зображення в 2-3 основних популярних форматах (bmp, png, jpg), мати панель інструментів для створення графічних примітивів, вибору кольорів, нанесення тексту, додавання найпростіших візуальних ефектів (ч/б растр, інфрачервоний растр, 2-3 на вибір учня), роботи з шарами.

Завдання.

- Ознайомитися з короткими теоретичними відомостями.
- Реалізувати частину функціонала робочої програми у вигляді класів і їх взаємодій для досягнення конкретних функціональних можливостей.
- Застосування одного з даних шаблонів при реалізації програми.

Виконання:

<https://github.com/Hrystynkkaa/trpz/tree/main/%D0%BA%D0%BE%D0%B4/GraphicEditor>

Flyweight :

ColorFactory оптимізує створення кольорів. Замість створення нового об'єкта Color щоразу, програма повторно використовує вже створені кольори з кешу, якщо вони мають ті самі параметри

```
public class ColorFactory {
    private static final Map<String, Color> colorCache = new HashMap<>();

    // Повертає колір з кешу або створює новий, якщо його немає
    public static Color getColor(int red, int green, int blue) {
        String key = red + "," + green + "," + blue; // Унікальний ключ для кожного
        кольору
        if (!colorCache.containsKey(key)) {
            colorCache.put(key, new Color(red, green, blue));
        }
        return colorCache.get(key);
    }

    public static int getCacheSize() {
        return colorCache.size();
    }
}
```

Висновок: У ході виконання роботи був застосований шаблон проектування Flyweight для оптимізації використання ресурсів при роботі з кольорами. Реалізація класу ColorFactory забезпечує збереження унікальних об'єктів кольорів у кеші, дозволяючи

повторно використовувати їх замість створення нових екземплярів. Це значно зменшує витрати пам'яті та покращує продуктивність у випадках, коли однакові кольори використовуються багаторазово.