



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 №7

із дисципліни «Технології розроблення програмного забезпечення»

Тема: «ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE METHOD»

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>

Bridge:

ImageSaver — це абстрактний клас, який визначає загальний інтерфейс для збереження зображень. У класі є посилання на об'єкт SaveImplementation, який відповідає за конкретну реалізацію збереження.

```
public abstract class ImageSaver {  
    protected SaveImplementation saveImplementation;  
  
    public ImageSaver(SaveImplementation saveImplementation) {  
        this.saveImplementation = saveImplementation;  
    }  
  
    public abstract void save(String filePath, BufferedImage image);  
}
```

ConcreteImageSaver — конкретна реалізація класу ImageSaver. Він просто делегує збереження зображення об'єкту saveImplementation, який визначає, як саме це буде зроблено (який формат, метод збереження і т.д.).

```
import java.awt.image.BufferedImage;
```

```

public class ConcreteImageSaver extends ImageSaver {
    public ConcreteImageSaver(SaveImplementation saveImplementation) {
        super(saveImplementation);
    }

    @Override
    public void save(String filePath, BufferedImage image) {
        saveImplementation.saveImage(filePath, image);
    }
}

```

SaveImplementation — це інтерфейс, який визначає метод збереження зображення. Клас **ImageSaver** використовує його для делегування збереження.

```

public interface SaveImplementation {
    void saveImage(String filePath, BufferedImage image);
}

```

RasterSaveImplementation і **VectorSaveImplementation** — конкретні реалізації інтерфейсу **SaveImplementation**, які відповідають за збереження зображень у різних форматах

```

public class VectorSaveImplementation implements SaveImplementation {
    @Override
    public void saveImage(String filePath, BufferedImage image) {
        File file = new File(filePath);
        try (FileWriter writer = new FileWriter(file)) {
            writer.write("<svg width=\"" + image.getWidth() + "\"" height=\"" +
image.getHeight() + "\"" xmlns=\"http://www.w3.org/2000/svg\">\n");
            writer.write("<rect width=\"100%\" height=\"100%\" fill=\"white\" />\n");
            writer.write("<!-- Векторні примітиви повинні бути додані тут -->\n");
            writer.write("</svg>");
            System.out.println("Векторне зображення збережено у форматі SVG: " +
filePath);
        } catch (IOException e) {
            System.err.println("Помилка збереження векторного зображення: " +
e.getMessage());
        }
    }
}

```

```

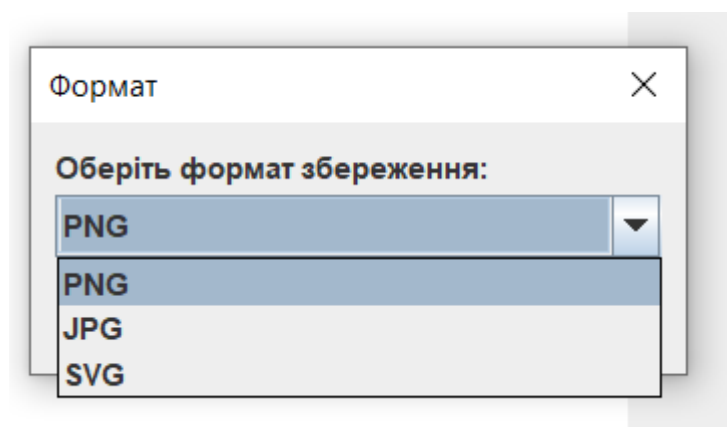
public class RasterSaveImplementation implements SaveImplementation {
    private final String format;

    public RasterSaveImplementation(String format) {
        this.format = format; // Формат: "png", "jpg", "bmp"
    }

    @Override
    public void saveImage(String filePath, BufferedImage image) {
        try {
            ImageIO.write(image, format, new File(filePath));
            System.out.println("Растрове зображення збережено у форматі " + format + ": "
+ filePath);
        } catch (IOException e) {
            System.err.println("Помилка збереження растрового зображення: " +
e.getMessage());
        }
    }
}

```

Результат:



Растрове зображення збережено у форматі png: D:\bridge

Висновок : У результаті виконання лабораторної роботи було реалізовано графічний редактор з використанням шаблону проектування Bridge. Цей шаблон дозволяє розділити абстракцію збереження зображень від конкретних реалізацій методів збереження. У нашому випадку, клас `ImageSaver` є абстракцією, яка делегує виконання збереження зображень класам, що реалізують інтерфейс `SaveImplementation`. Класи `RasterSaveImplementation` та `VectorSaveImplementation` надають конкретні реалізації для збереження в растрових та векторних форматах відповідно.

Завдяки такому підходу ми можемо легко додавати нові формати зображень або змінювати спосіб їх збереження, не впливаючи на основний код програми. Крім того, це дозволяє підтримувати принцип відкритості/закритості, де система відкрита для розширень, але замкнена для змін.

Це дозволяє значно зменшити складність коду та покращити його підтримуваність, оскільки нові функціональні можливості можна додавати без необхідності переписувати існуючі класи.