



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 №9

із дисципліни «Технології розроблення програмного забезпечення»

Тема: «РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ: CLIENT-SERVER, PEER-TO-PEER,
SERVICE-ORIENTED ARCHITECTURE»

Перевірив:
Мякий 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 функціоналу робочої програми у вигляді класів і їх взаємодій для досягнення конкретних функціональних можливостей.
- Реалізувати взаємодію програми в одній з архітектур відповідно до обраної теми.

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

Шаблон SOA:

1. Сервіси

ColorService: керує кольорами.

ImageService: працює із зображеннями

LayerService: керує шарами в графічному редакторі.

Ізоляція логіки: Кожен сервіс відповідає за свою вузьку область (кольори, зображення, шари).

Повторне використання: Ці сервіси можна використовувати в різних частинах програми, наприклад, у графічному інтерфейсі.

2. Репозиторії для роботи з даними

Пакет repository відповідає за доступ до даних:

ColorRepository та його реалізація ColorRepositoryImpl керують збереженням і отриманням даних про кольори.

ImageRepository працює із зображеннями.

LayerRepository керує даними про шари.

3. Моделі як спільний обмін даними

У пакеті model:

Класи, що представляють доменні об'єкти: Color, Image, Layer, Circle, Rectangle.

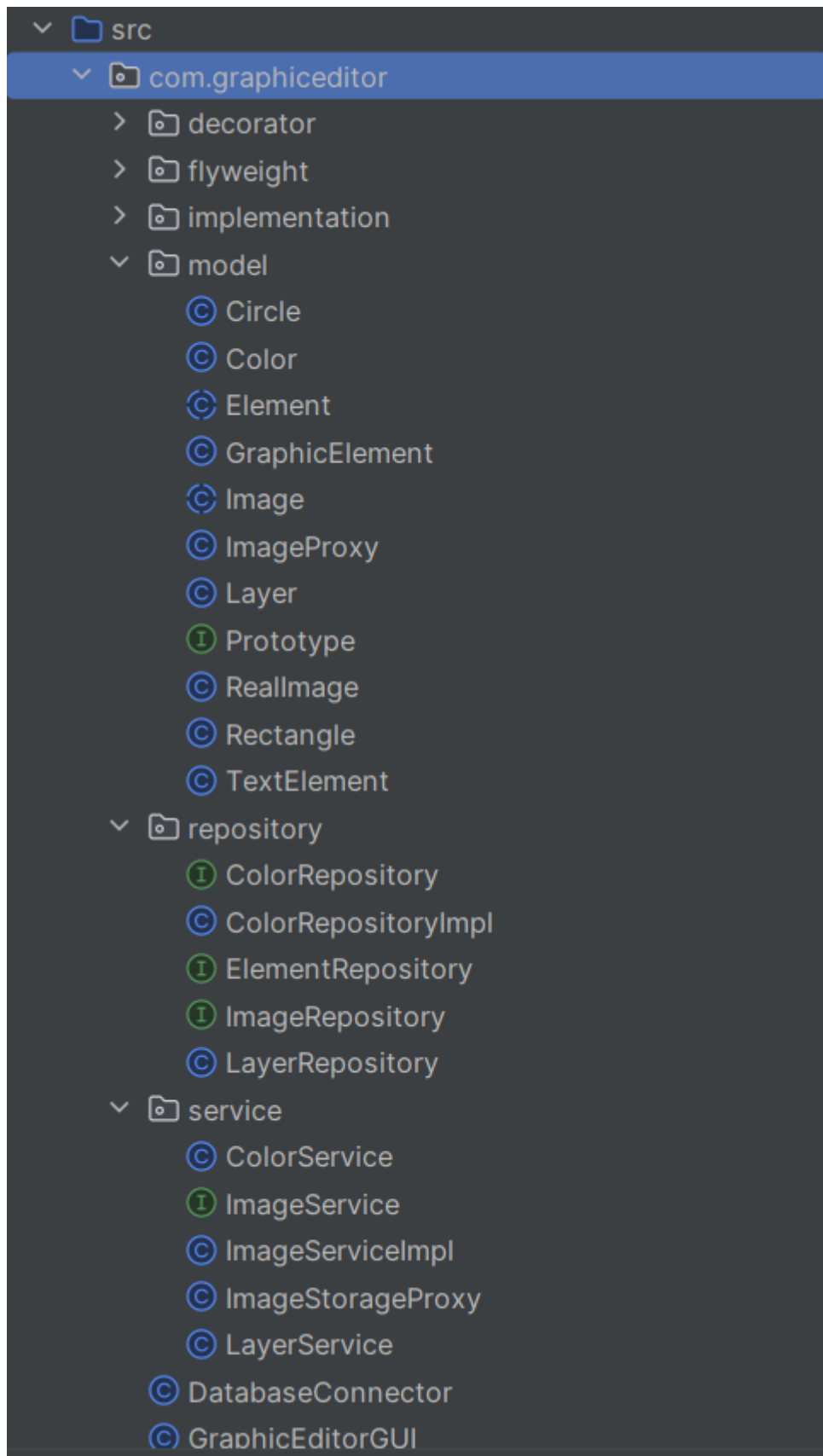
Додаткові шаблони: Prototype, ImageProху.

4. Фасад або централізований доступ до сервісів

Клас GraphicEditorGUI виступає як фасад для доступу до всіх сервісів. Він:

Використовує сервіси (ColorService, ImageService) для виконання дій.

Є єдиною точкою доступу для графічного інтерфейсу.



Висновок: Під час виконання даної лабораторної роботи було реалізовано ключові принципи сервіс-орієнтованої архітектури (SOA) у структурі програми. Було створено систему, яка забезпечує чіткий розподіл обов'язків між компонентами та сприяє легкому розширенню і модифікації. Зокрема:

Модульність: Було виокремлено сервіси (ColorService, ImageService, LayerService), які відповідають за різні аспекти функціонування програми. Це дозволяє уникнути дублювання коду та підвищити зрозумілість програми.

Використання шаблонів проектування: Реалізація шаблонів Flyweight і Decorator допомогла оптимізувати споживання ресурсів та забезпечити гнучкість у роботі із графічними елементами. Зокрема, Flyweight дозволив знизити кількість об'єктів у пам'яті через використання кешування кольорів.

Чіткий розподіл на рівні даних: Репозиторії (ColorRepository, ImageRepository) забезпечили зручний і незалежний доступ до даних, що дозволяє змінювати джерела даних без впливу на бізнес-логіку.

Гнучкість та масштабованість: Завдяки сервіс-орієнтованій архітектурі систему легко адаптувати до змін або розширювати функціонал шляхом додавання нових сервісів чи модифікації існуючих.

Моделі даних: Створені класи моделей (Color, Image, Layer, Rectangle, Circle) забезпечують зручну роботу із графічними елементами та служать основою для їх використання в сервісах.