

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Звіт

з лабораторної роботи № 5

з дисципліни «Ефективність та якість архітектурних рішень
інформаційних систем»

Виконав:

студент групи ІКМ-М225В

Суліма Нікіта Володимирович

Перевірив:

аспірант каф. КМПС Хорошун Андрій Сергійович

Харків 2025

Інтерфейс Renderer

```
class Renderer {  
public:  
    virtual string renderText(string text) = 0;  
    virtual string renderImage(string url) = 0;  
    virtual string renderLink(string url, string title) = 0;  
};
```

Конкретні реалізації Renderer

- HTMLRenderer — повертає HTML-код.
- JsonRenderer — повертає дані у форматі JSON.
- XmlRenderer — повертає дані у форматі XML.

```
string renderText(string text) override {  
  
    return "<p>" + text + "</p>";  
  
}
```

Абстракція Page

```
class Page {  
protected:  
    Renderer* renderer;  
public:  
    Page(Renderer* r) : renderer(r) {}  
    virtual string view() = 0;  
};
```

Page — це абстракція, яка використовує об'єкт типу Renderer для створення сторінки.

Абстракція не знає, в якому форматі буде відображення — HTML, JSON чи XML.

Refined Abstractions (спеціалізації сторінок)

SimplePage

```
class SimplePage : public Page {  
    string title, content;  
    string view() override {  
        return renderer->renderText(title) + "\n" +
```

```
        renderer->renderText(content);  
    }  
};
```

ProductPage

```
class ProductPage : public Page {  
    Product* product;  
    string view() override {  
        return renderer->renderText(product->name) + "\n" +  
            renderer->renderText(product->description) + "\n" +  
            renderer->renderImage(product->image) + "\n" +  
            renderer->renderLink("/product/" + product->id, "View Product");  
    }  
};
```

Клас Product

Зберігає інформацію про товар:

```
class Product {  
public:  
    string id, name, description, image;  
};
```

Клієнтський код

```
int main() {  
    Renderer* html = new HTMLRenderer();  
    Renderer* json = new JsonRenderer();  
    Renderer* xml = new XmlRenderer();  
  
    Page* simple = new SimplePage(html, "About Us", "Welcome to our site!");  
    cout << simple->view();  
  
    Product* product = new Product("101", "Laptop", "Powerful gaming laptop",  
    "laptop.jpg");  
    Page* productPageXml = new ProductPage(xml, product);  
    cout << productPageXml->view();  
}
```

Висновок

У даній лабораторній роботі реалізовано патерн «Міст» (Bridge), який дозволяє розділити логіку побудови сторінок (Page) від логіки рендерингу,

легко додавати нові формати виводу, повторно використовувати абстракції без переписування коду. Таким чином, програма демонструє головну ідею патерну — незалежність абстракції від реалізації.