

## Лабораторна робота №3 (максимально - 10 балів)

### Тема: Структурні шаблони

**Мета роботи:** навчитися реалізовувати структурні шаблони проєктування *Адаптер, Декоратор, Міст, Компонувальник, Проксі, Легковаговик*

### Завдання на лабораторну роботу

#### Завдання 0: Підготовка до виконання завдання

1. Створити директорію lab-3 в Вашому репозиторії з лабораторками.
2. Перейти до Завдання 1 😊

#### Завдання 1: Адаптер.

1. Створіть клас Logger, який буде мати методи `Log()`, `Error()`, `Warn()`, які виводять повідомлення в консоль різними кольорами (зеленим, червоним і оранжевим відповідно).
2. Створіть клас FileWriter з методами `Write()`, `WriteLine()`.
3. За допомогою шаблону Адаптер створіть файловий логер.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми

#### Завдання 2: Декоратор.

1. Ви розробляєте РПГ гру. Створіть класи героїв Warrior, Mage, Paladin.
2. Для героїв створіть інвентар (одяг, зброю, артефакти), який може підходити будь-якому типу героїв, у вигляді декораторів.
3. Важливою вимогою є можливість використання декількох екземплярів інвентаря на герої одночасно.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

#### Завдання 3: Міст.

1. Ви працюєте над графічним редактором. Створіть базовий клас Shape.

2. Створіть дочірні до Shape класи, Circle, Square, Triangle.
3. За допомогою шаблону Міст додайте можливість рендерингу кожної з фігур як векторної або растрової графіки (вивівши відповідне повідомлення у консоль, наприклад "Drawing Triangle as pixels").
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

#### Завдання 4: Проксі.

1. Створіть клас SmartTextReader, який вміє читати вміст текстового файлу і перетворювати його на двовірний масив якому зовнішній масив відповідає рядкам тексту, а вкладені масиви відповідають символам у відповідному рядку.
2. Створіть проксі для SmartTextReader з логуванням SmartTextChecker, який буде виводити інформацію про успішне відкриття, прочитання і закриття файлу, а також буде виводити загальну кількість рядків і символів у прочитаному тексті.
3. Створіть проксі для SmartTextReader з обмеженням доступу до певних файлів SmartTextReaderLocker. Цей клас в конструкторі приймає регулярний вираз, по якому лімітується доступ до певної групи файлів. Якщо клієнт викликатиме метод для прочитання такого лімітованого файлу, замість прочитання файлу в консоль має виводитися повідомлення "Access denied!".
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

#### Завдання 5: Компонувальник.

1. Вам потрібно створити власну мову розмітки LightHTML.
2. Кожен елемент розмітки має наслідувати клас LightNode.
3. Створіть два дочірніх класи від LightNode: LightElementNode, LightTextNode.
4. LightTextNode може містити лише текст.
5. LightElementNode може містити будь-які LightNode. LightElementNode повинен мати інформацію про назву тега, його тип відображення (блочний чи рядковий), тип закриття (одиничний тег, як `<img/>` чи з закриваючим тегом) список CSS класів,

кількість дочірніх елементів, а також має бути можливість виводити на екран його outerHTML і innerHTML.

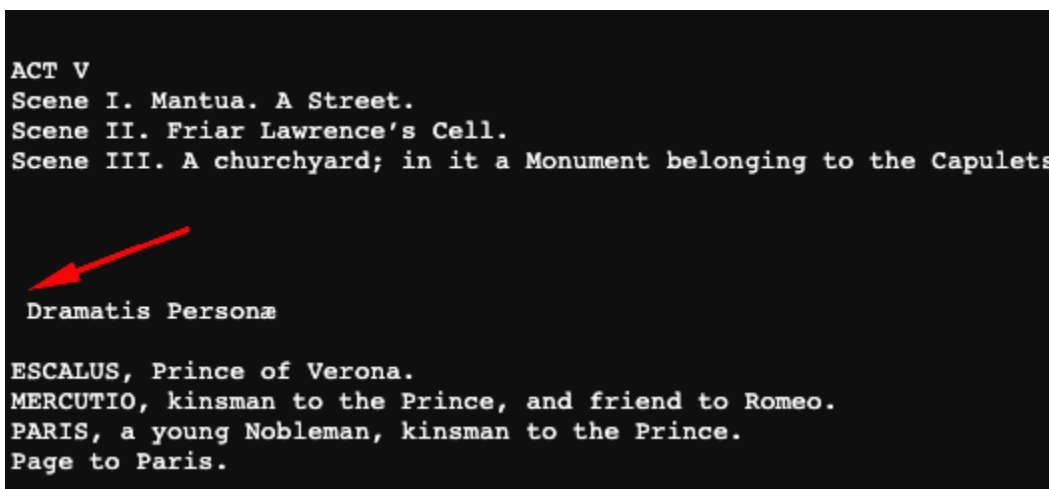
6. За допомогою своєї мови розмітки виведіть в консоль елемент сторінки на Ваш вибір (наприклад якусь таблицю, список тощо).

7. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

### Завдання 6: Легковаговик.

1. За допомогою свого LightHTML з завдання 1 перетворіть текст [книги](#) в HTML верстку за такими правилами:

- Перший рядок має бути елементом `<h1>`
- Якщо в рядку менше 20 символів - це має бути елемент `<h2>`
- Якщо рядок починається з пробільного символу - це має бути `<blockquote>`.



```
ACT V
Scene I. Mantua. A Street.
Scene II. Friar Lawrence's Cell.
Scene III. A churchyard; in it a Monument belonging to the Capulets

Dramatis Personæ

ESCALUS, Prince of Verona.
MERCUTIO, kinsman to the Prince, and friend to Romeo.
PARIS, a young Nobleman, kinsman to the Prince.
Page to Paris.
```

- В будь-якому іншому випадку - елемент `<p>`

2. Покажіть скільки займає все дерево Вашої верстки, коли воно повністю утримується в пам'яті процесу.

3. Використайте Легковаговик на Ваших класах HTML елементів, щоб зменшити споживання пам'яті.

4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.