

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

Лабораторна робота №8
із дисципліни «*Технології розробки програмного забезпечення*»
Тема: «*Патерни проектування*»

Виконав:

Студент групи ІА-34
Ястремський Богдан

Перевірив:

асистент кафедри ІСТ
Мягкий Михайло Юрійович

Тема: Патерни проектування.

Мета: Вивчити структуру шаблонів «Composite», «Flyweight» (Пристосуванець), «Interpreter», «Visitor» та навчитися застосовувати їх в реалізації програмної системи.

Застосунок (№6): Web-browser (proxy, chain of responsibility, factory method, template method, visitor, p2p).

Веб-браузер повинен мати можливість зробити наступне: мати адресний рядок для введення адреси сайту, переміщатися і відображати структур html документа, переглядати підключений javascript та css файли, перегляд всіх підключених ресурсів (зображень), коректна обробка відповідей з сервера (коди відповідей HTTP) – переходи при перенаправленнях, відображення сторінок 404 і 502/503.

Короткі теоретичні відомості:

Шаблон «Компонувальник» (Composite) призначений для складання об'єктів у деревоподібну структуру, що дозволяє представити ієрархії типу «частина-ціле». Основна перевага цього шаблону полягає в уніфікації обробки: він дає змогу працювати як з поодинокими об'єктами (листами), так і з об'єктами, що містять вкладені елементи (композицями), використовуючи при цьому єдиний інтерфейс. Цей патерн є найбільш зручним і доцільним рішенням, коли виникає необхідність подання та ефективної обробки ієрархічних структур даних, типовим прикладом яких є дерево.

Шаблон «Flyweight» використовується для зменшення кількості об'єктів в додатку шляхом поділу цих об'єктів між ділянками додатку. Flyweight являє собою поділюваний об'єкт. Дуже важливою є концепція «внутрішнього» і «зовнішнього» станів. Внутрішній стан відображає дані, характерні саме поділюваному об'єкту (наприклад, код букви); зовнішній стан несе інформацію про його застосування в додатку (наприклад, рядок і стовпчик). Внутрішній стан зберігається в самому поділюваному об'єкті, зовнішній – в об'єктах додатку (контексту використання поділюваного об'єкта).

Шаблон «Інтерпретатор» визначає граматику та інтерпретатор для мови, наприклад, скриптової. Граматика описується термінальними і нетермінальними символами. Клієнт передає контекст і речення у вигляді абстрактного синтаксичного дерева. Кожен вираз в АСД інтерпретується окремо з використанням контексту. Батьківський вираз інтерпретує спочатку всі дочірні вирази (рекурсивно), а потім обчислює власний результат.

Шаблон «Відвідувач» дозволяє додавати нові операції над елементами ієрархії без зміни їхньої структури. Це спрощує додавання нових операцій (нових Відвідувачів), але ускладнює додавання нових елементів до ієрархії (потрібно оновлювати всіх Відвідувачів). Шаблон Відвідувач ефективно групує однотипні операції, що застосовуються до об'єктів різних типів.

Хід роботи:

1. Повна діаграма класів (без шаблону).

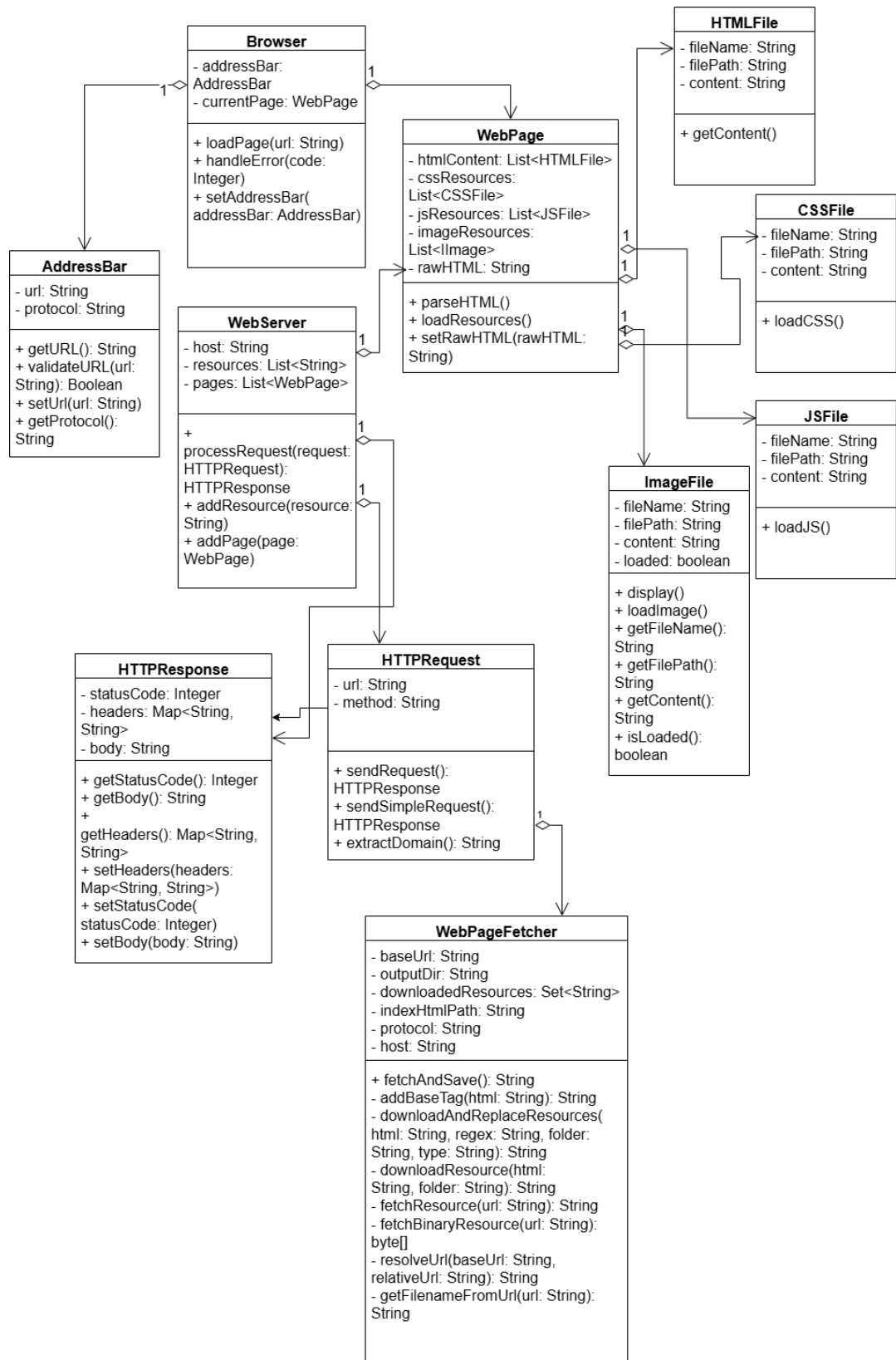


Рис. 8.1 – Діаграма класів системи

2. Діаграма класів з використанням шаблону «Visitor»:

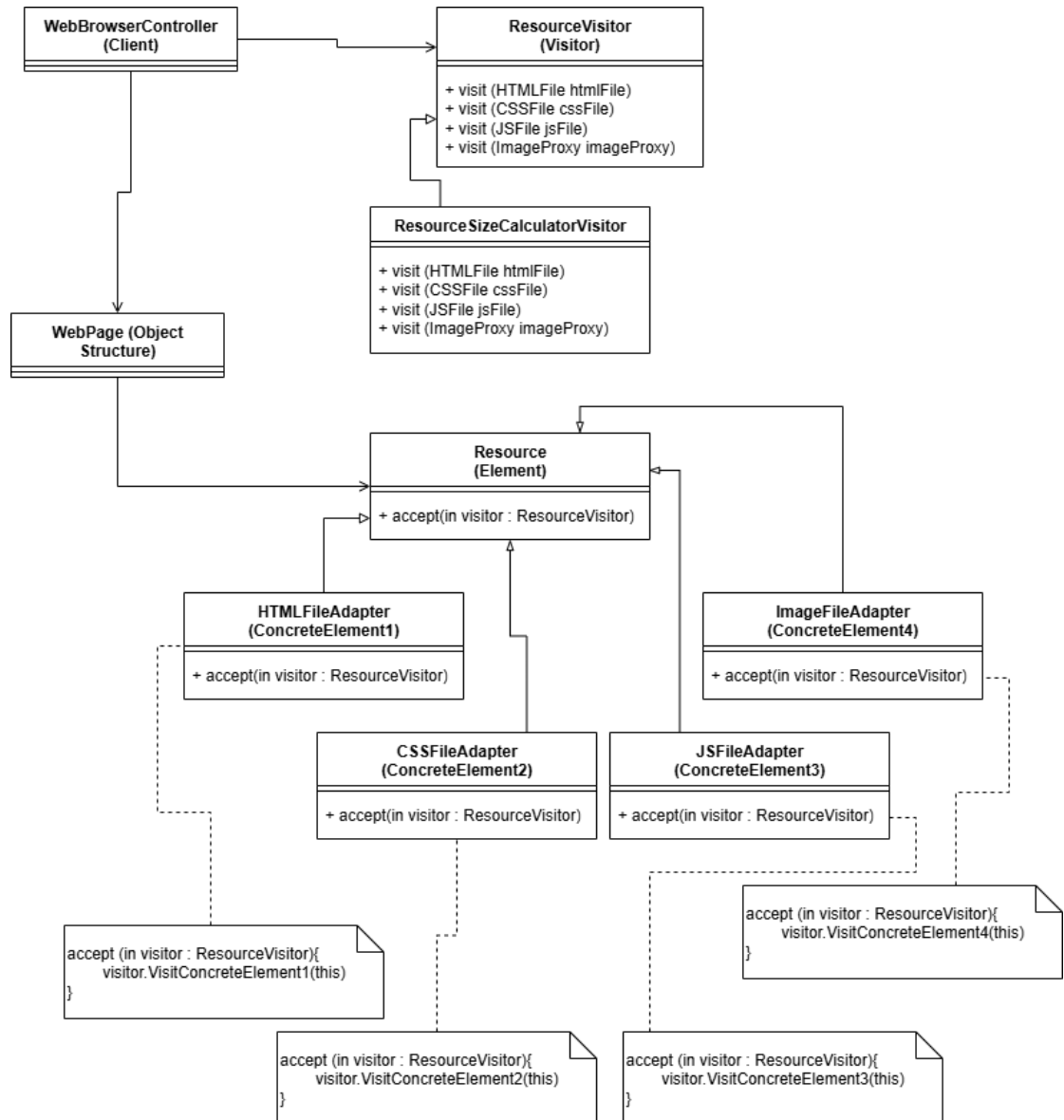


Рис. 8.2 – Діаграма класів системи з використанням шаблону «Visitor»

Я визначив задачу статистичного збору розміру завантажуваних ресурсів. Першим ділом, я визначив **WebPage** та контролер як **Object Structure** та **Client** відповідно, оскільки вони клієнт ініціює використання патерну, а структура об'єкту визначає метод для обходу ресурсів.

Далі я оголосив інтерфейс **ResourceVisitor**, який визначає методи візиту для кожного ресурсу – `html`, `css`, `js`, `img` із подальшим створенням класу **ResourceSizeCalculatorVisitor**, який реалізує ці методи і збирає статистичні дані.

Далі, як елемент, я оновив інтерфейс Resource, в який додав оголошення методу accept().

Фінальним кроком стало оновлення класів HTMLFileAdapter, CSSFileAdapter, JSFileAdapter, ImageFileAdapter, які реалізують метод accept() та самих себе (this) visitor-у.

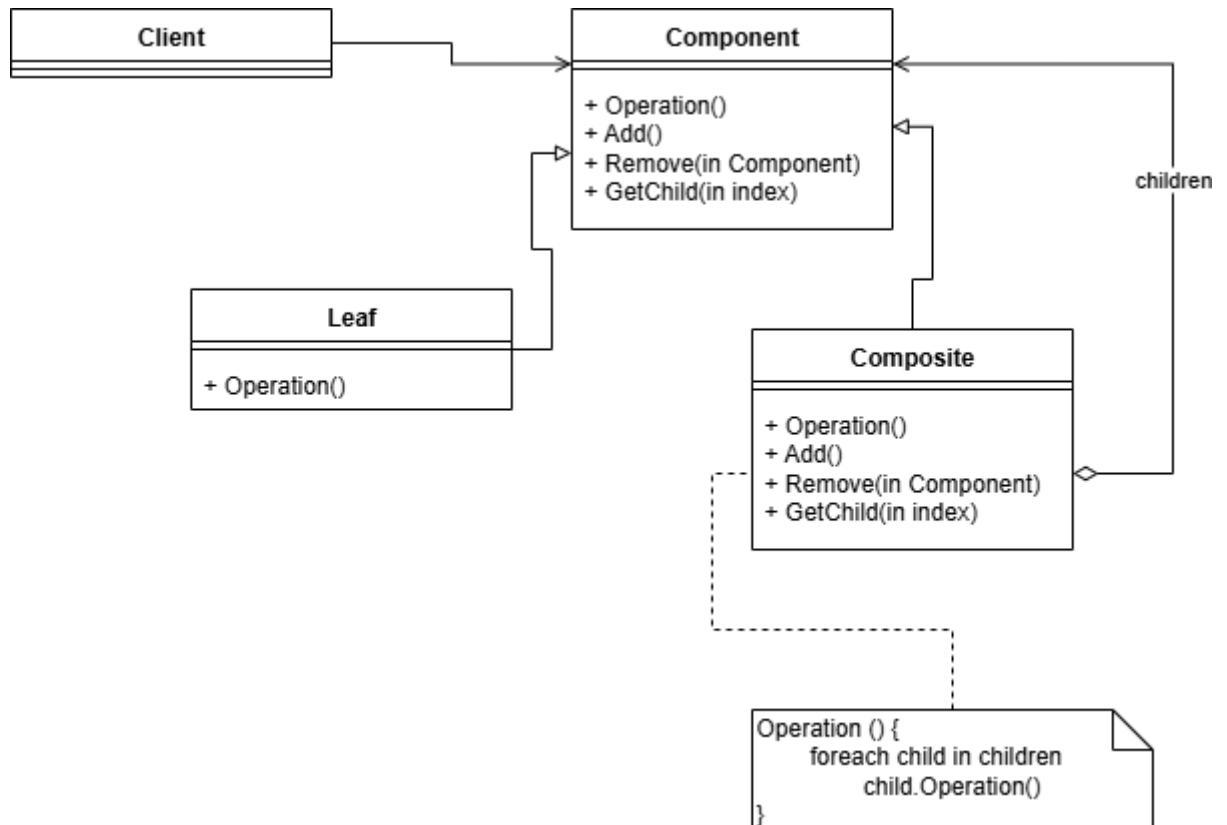
Висновок: на даному лабораторному занятті я продовжив знайомством з поняттям патернів проектування, які є їх види, та їхні плюси і мінуси. В якості використовуваного патерну я обрав «Visitor», та імплементував його як збирач статистичних даних завантажуваної сторінки – загальний розмір всіх видів кожного ресурсу. Таким чином, я зміг спростити додавання нових функцій, що працюють зі складною структурою об'єктів, без зміни коду самих цих об'єктів. Більш того, об'єкт структури відповідає лише за свою будову та можливість "прийняти відвідувача", а алгоритми обробки знаходяться окремо, що є гарним прикладом відділення алгоритму від структури.

Відповіді на контрольні питання:

1. Яке призначення шаблону «Композит»?

Шаблон використовується для складання об'єктів в деревоподібну структуру для подання ієрархій типу «частина цілого». Даний шаблон дозволяє уніфіковано обробляти як поодинокі об'єкти, так і об'єкти з вкладеністю.

2. Нарисуйте структуру шаблону «Композит».



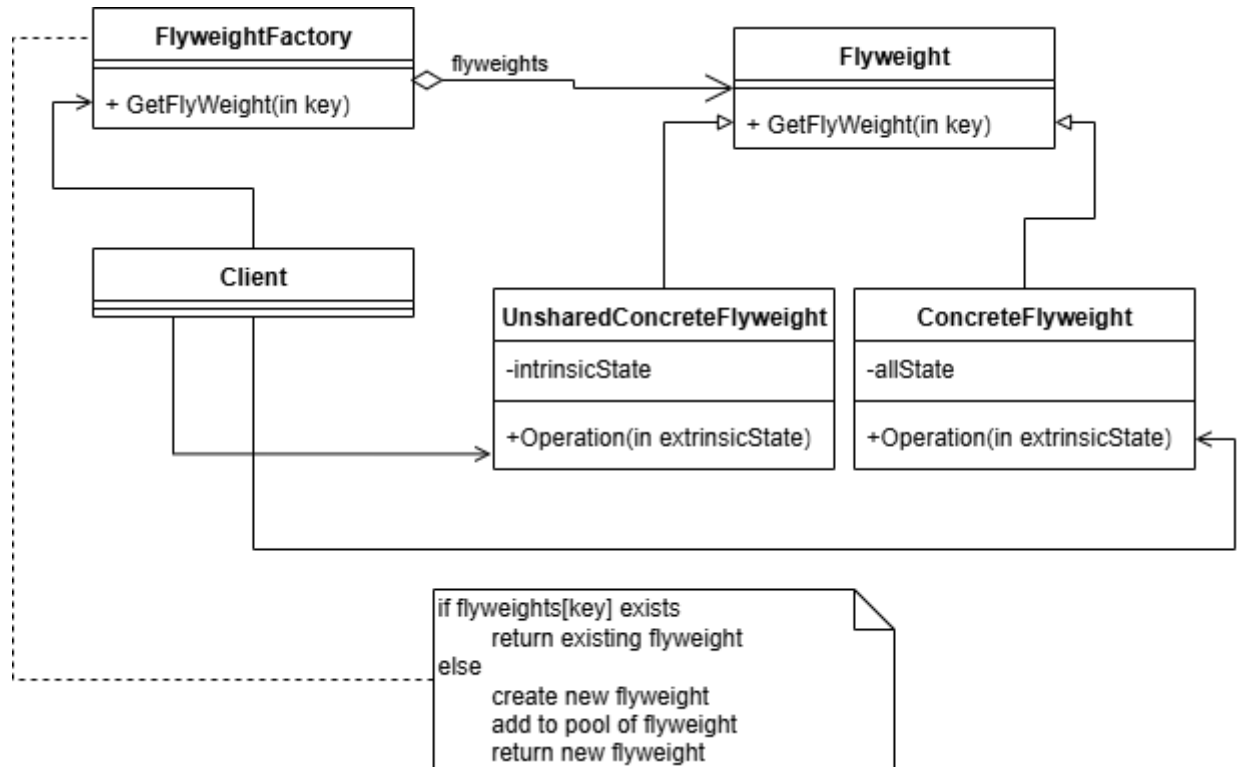
3. Які класи входять в шаблон «Композит», та яка між ними взаємодія?

- **Component**: Це абстрактний базовий клас або інтерфейс. Він визначає загальний інтерфейс для всіх об'єктів у структурі (як простих, так і складних).
- **Leaf**: Представляє кінцеві, прості об'єкти структури, які не можуть містити дочірні елементи.
- **Composite**: Представляє складні об'єкти, які можуть містити інші Компоненти.

4. Яке призначення шаблону «Легковаговик»?

Шаблон використовується для зменшення кількості об'єктів в додатку шляхом поділу цих об'єктів між ділянками додатку. Flyweight являє собою поділюваний об'єкт.

5. Нарисуйте структуру шаблону «Легковаговик».



6. Які класи входять в шаблон «Легковаговик», та яка між ними взаємодія?

- Flyweight: Визначає інтерфейс, через який "легковаговики" можуть отримувати та використовувати свій зовнішній (контекстний) стан.
- Concrete Flyweight: Реалізує інтерфейс Легковаговика і відповідає за зберігання внутрішнього (intrinsic) стану — даних, які є спільними (незмінними) для багатьох об'єктів і можуть бути розділені.
- Flyweight Factory: Керує створенням та спільним використанням (пулом) об'єктів Легковаговиків.
- Client: Створює об'єкти Легковаговиків через Фабрику і обчислює або зберігає зовнішній (extrinsic) стан — унікальні дані, які не можна розділити між об'єктами.

7. Яке призначення шаблону «Інтерпретатор»?

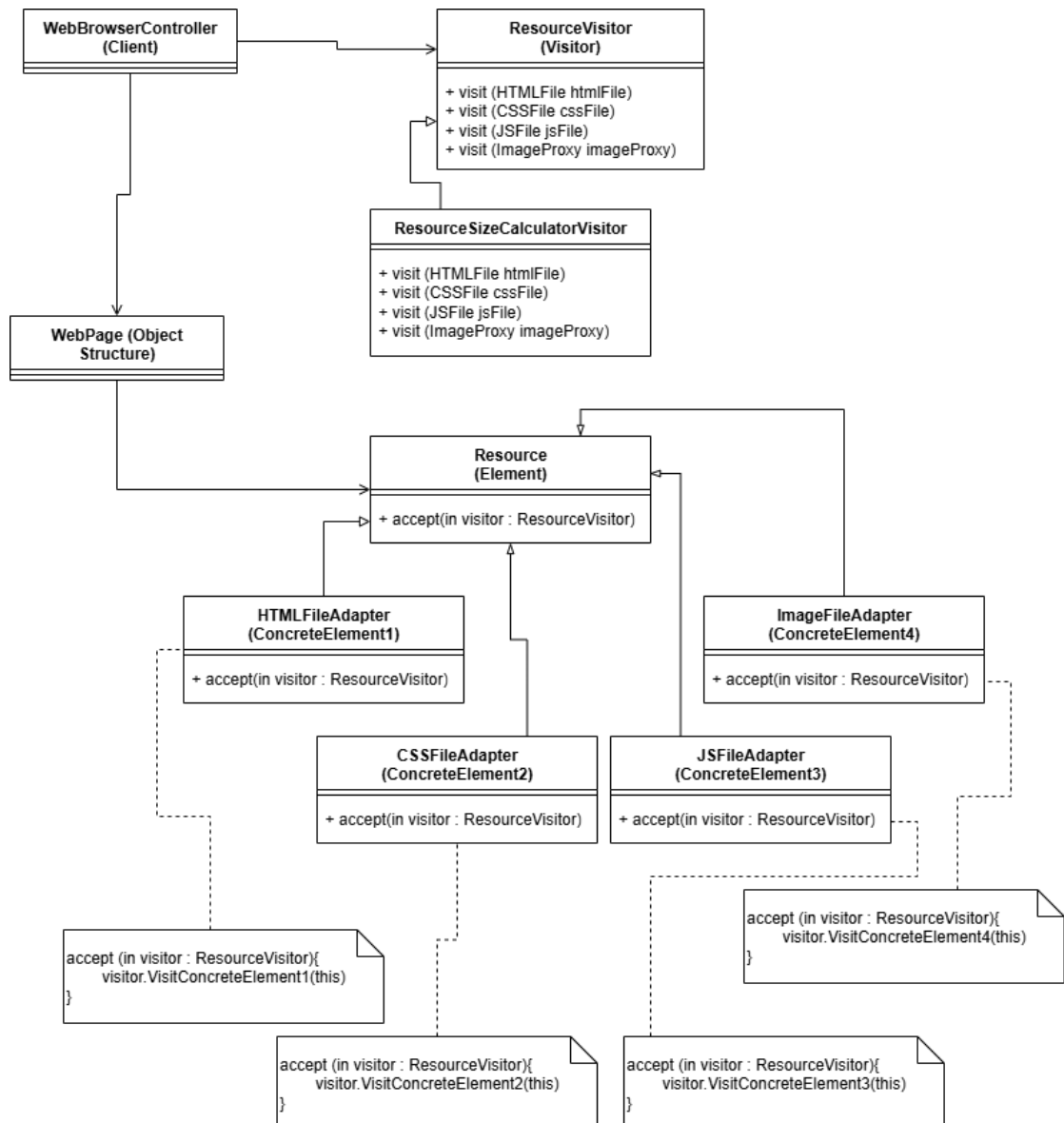
Шаблон «Інтерпретатор» визначає граматику та інтерпретатор для мови, наприклад, скриптової. Граматика описується термінальними і

нетермінальними символами. Клієнт передає контекст і речення у вигляді абстрактного синтаксичного дерева. Кожен вираз в АСД інтерпретується окремо з використанням контексту. Батьківський вираз інтерпретує спочатку всі дочірні вирази (рекурсивно), а потім обчислює власний результат.

8. Яке призначення шаблону «Відвідувач»?

Шаблон відвідувач дозволяє вказувати операції над елементами без зміни структури конкретних елементів. Таким чином вкрай зручно додавати нові операції. Даний шаблон дозволяє групувати однотипні операції, що застосовуються над різнотипними об'єктами.

9. Нарисуйте структуру шаблону «Відвідувач».



10. Які класи входять в шаблон «Відвідувач», та яка між ними взаємодія?