

[Visitor Design Pattern]

The **Visitor Pattern** is a behavioral design pattern that separates an algorithm from the objects it operates on. It allows you to add new operations to a group of classes without modifying their structure. This is achieved by creating a visitor object that implements the desired operations, and this visitor object is passed to the elements it will operate on.

Key Concepts:

1. Visitor Interface:

- Declares methods for each type of object that can be visited.

2. Concrete Visitor:

- Implements the operations to be performed on the elements.

3. Element Interface:

- Declares an `accept` method that takes a visitor object.

4. Concrete Elements:

- Implement the `accept` method to call the visitor's method corresponding to their type.

5. Client:

- Traverses elements and passes the visitor to them.