Visitor

Objective

Represent an operation to be performed on the elements of an object structure; allowing to define a new operation without changing the classes of the operating elements.

Function

Representing a function to be performed by the elements of the object's structure, it also allows defining a new function without changing the classes of the elements in which it operates.

Structure

As shown in figure 1

- Client: Component that interacts with the structure (element) and with the Visitor. The Visitor is responsible for creating the visitors and sending them to the element for processing.

- Element: Represents the root of the structure, in the form of a tree, on which we will use the Visitor. This object is usually an interface that defines the accept method and should implement all the objects of the structure.

- ElementConcrete (A,B): Represents a child of the composite structure, the complete structure can be composed of a large number of these objects and each must implement the accept method.

- IVisitor: Interface that defines the structure of the visitor, the interface should have a method for each object that requires analysis of the structure (element).

- VisitorConcrete (A,B): It represents an implementation of the visitor, this implementation can perform an operation on the element. It is possible to have all the ConcreteVisitor needed to perform the operations we need.

The structure that meets this pattern is shown in Figure 1

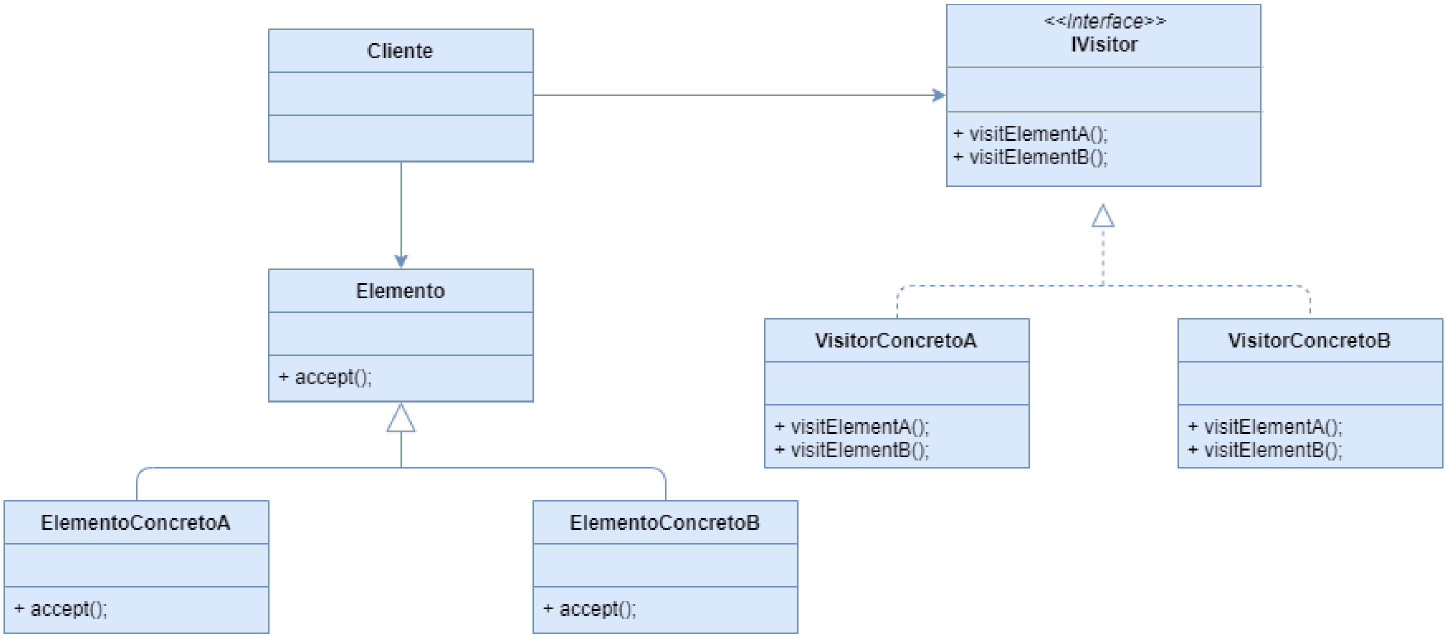

Figure 1: UML Diagram Visitor Pattern

Applications

The use of the Visitor pattern is recommended when:

- An object structure contains many classes of objects that differ in their interface and the aim is to combine the operation of their particular classes.
- Many different and unrelated operations need to be executed on objects and object structures; the Viewer object allows you to keep the related operation together by defining it in a class.
- Classes define object structures that rarely change; but highly changing requirements need to be defined.

Design Patterns Collaborators

- The Visitor pattern can be used to apply an operation on an object structure defined by a Composite pattern.
- The Visitor objects can be applied for the interpretation of Interpreter objects.

Scope of action

Applied at the object level.

Problem

The application requires many different operations that are not related, in the node object of an aggregated heterogeneous structure; taking into account that it cannot be allowed to fill the node class with this type of operations; increasing the complexity when analyzing each node until finding the appropriate one to launch the indicator to perform the desired operation.

Solution

The Visitor pattern is the object oriented model that allows the creation of an external class that acts on the data of the other classes, this process is highly useful; since it facilitates the use only of the instances of the classes that are needed to solve the needs of the moment without the need to involve or affect those that are not useful in a certain operation.

Diagram or Implementation

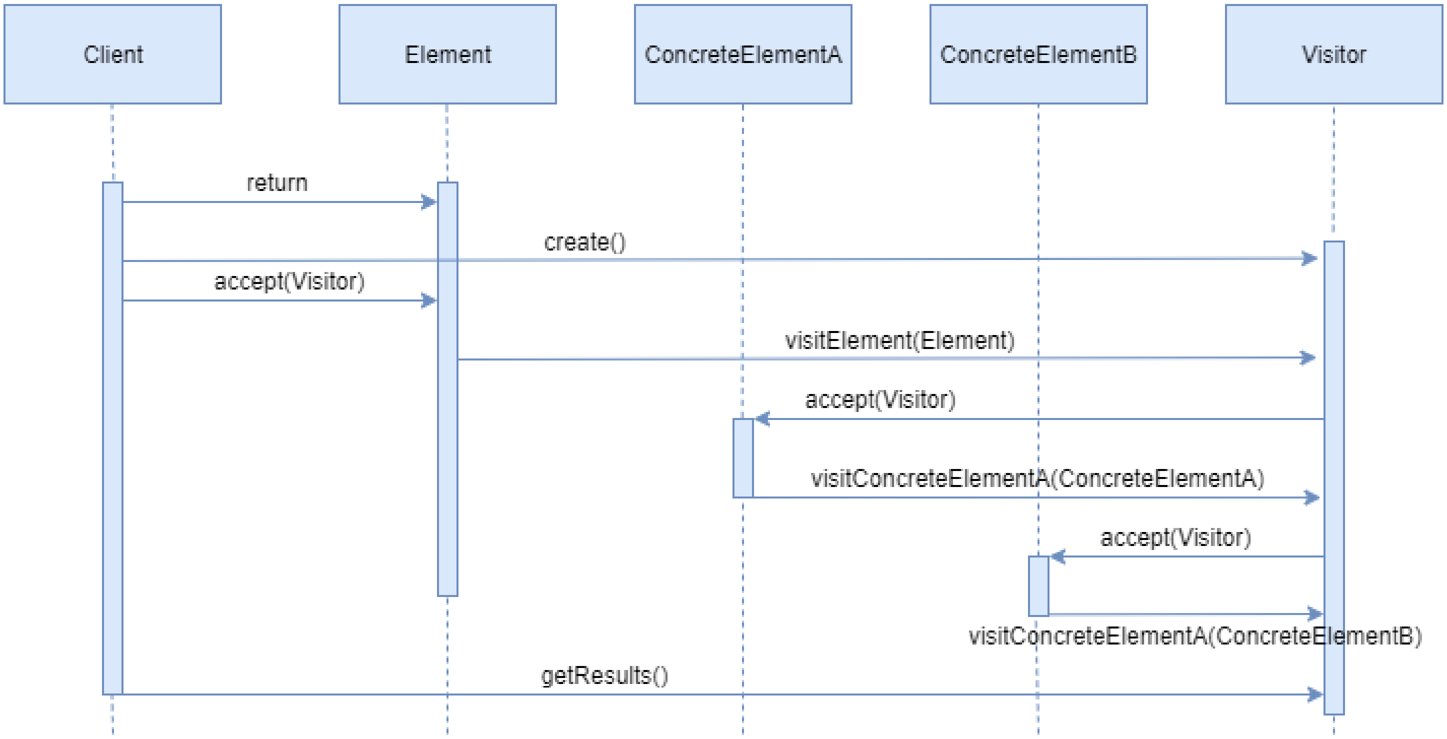

Figure 2: UML Diagram Visitor Pattern

Figure 2 explains the behaviour of the pattern by means of a sequence diagram.

- The client class creates the structure (Element).

- The client class creates the instance of the Visitor to be used on the structure.

- The client class executes the accept method of the structure and sends it to the Visitor.

- The Element component tells the Visitor which method to process it with. The Visitor must have a method for each class type in the structure.

- The Visitor analyzes the Element using its visitElement method and repeats the process of running the accept method on the children of the Element. Again the Visitor should have a method to process each child class of the structure.

- The ConcreteElementA component tells the Visitor which method to process it with, which is visitElementA.

- The visitor continues with the other children of Element and this time executes the accept method on the ConcreteElementB component.

- The ConcrteElementB tells the Viewer which method to process it with, which is visitElementB.

- Finally the Visitor finishes the operation on the structure when it has gone through all the objects, obtaining a result that is requested by the client class through the getResults method (the result is optional since there are operations that do not give results).