

Diseño y Arquitectura de Software

- Patrón de diseño Visitor

Alumno: Rubén Campos González

Nombre y clasificación

- Se clasifica dentro de los patrones de diseño de comportamiento. Estos buscan asignar responsabilidades y establecer una comunicación entre objetos.

Intención

- Usar el patrón visitante cuando quieras agregar funcionalidades a un grupo de objetos.
- Hacer la operación correcta basado en el tipo de dos objetos.

Motivación

- Representar una operación a ejecutar en los objetos. Definir una nueva operación sin cambiar las clases de los elementos con los que opera.
- De lo contrario, quien sabe que nuevos métodos se tengan que agregar despues la aplicación, y cada vez que se agregue un nuevo metodo se tendran que hacer en dos o mas lugares.
- Definir un objeto separado (visitor) que implemente una operación a ser ejecutada en los elementos de una estructura de objetos.
- Esto hace posible el crear nuevas operaciones independientemente de las clases de una estructura de objetos con solo añadir nuevos objetos de tipo visitante.

Aplicabilidad

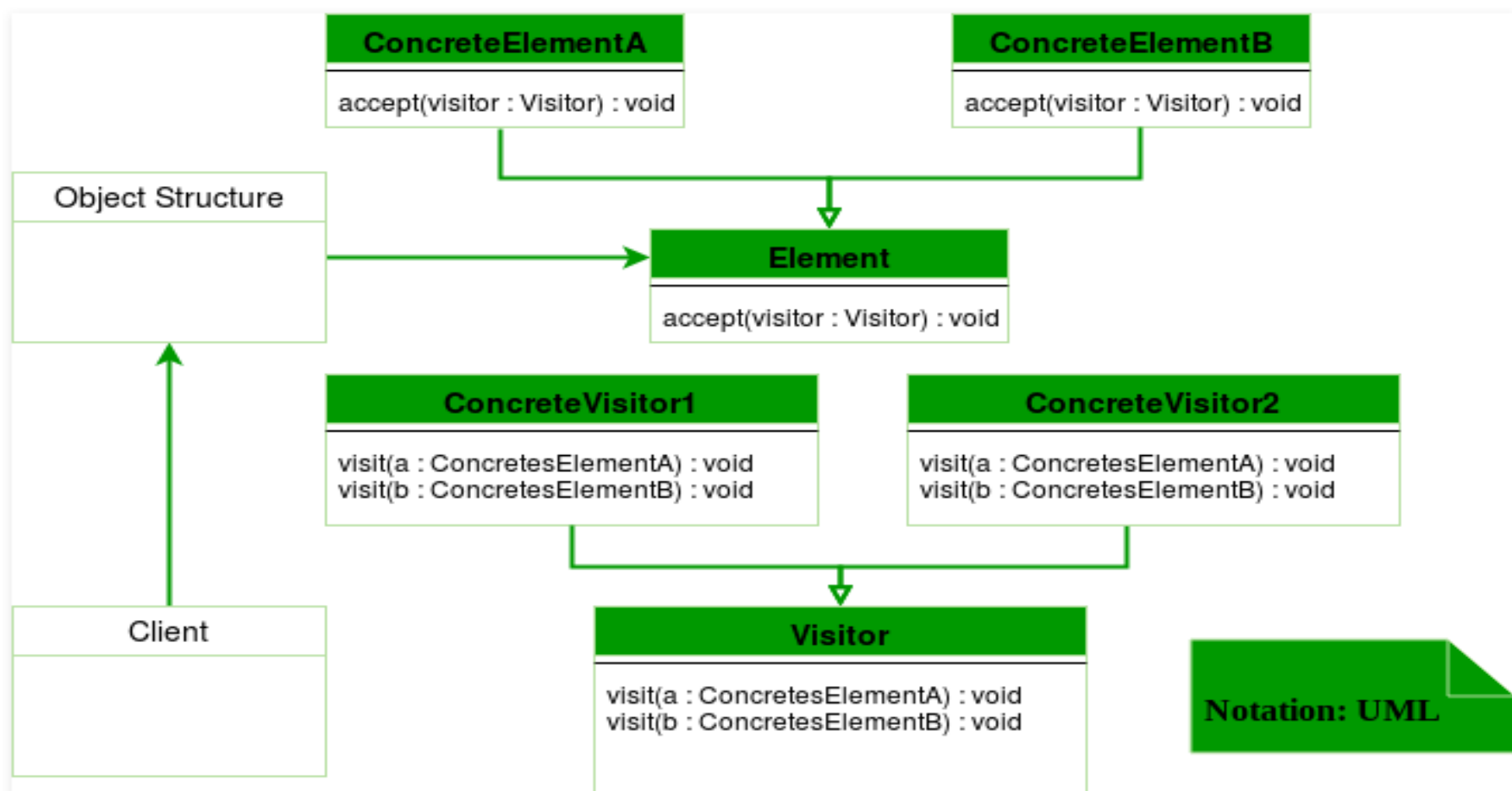
- El patrón visitante es aplicable, por ejemplo, cuando varias clases de objetos con interfaces diferentes y se desean realizar operaciones que dependen de sus clases concretas. También cuando se necesitan diversas operaciones sobre objetos de una jerarquía y no se desea recargar las clases con estas operaciones.
- Es de mucha utilidad cuando las clases de la jerarquía no cambian, pero se añaden con frecuencia operaciones a la estructura.
- Si la jerarquía cambia no es aplicable, ya que cada vez que agrega nuevas clases que deben ser visitadas, hay que añadir una operación “visita” abstracta a la clase abstracta del visitante, y debe agregar una aplicación de dicha categoría a cada Visitante concreto que se haya escrito.

Ejemplo Sencillo

- Método simple sin usar este patrón: supongamos un veterinario a domicilio, que con base al animal que tiene que curar sabe qué cura tiene que aplicar. Es responsabilidad del veterinario el saber de qué clase es el animal y sobre la base de ello aplicar la cura.
- Método usando el patrón: El veterinario no tiene que reconocer al animal ni saber qué cura aplicarle. Es el propio animal el que le dice al veterinario qué cura es más apropiada para él. El veterinario sólo "visita" al animal, y aplica la cura apropiada.

Estructura

UML Diagram Visitor design pattern



Participantes

- Visitor. Es una interfaz o una clase abstracta usada para declarar las funciones visit para todos los tipos de clases visitables. El tipo del objeto de entrada decide cual función es llamada.
- ConcreteVisitor. Por cada tipo de visitante, todas la funciones visit declaradas en la clase abstracta Visitor deben ser implementadas. Cada visitor sera responsable de diferentes operaciones.
- Element. Es una interfaz o una clase abstracta que declara la función accept. Éste es el punto de entrada que habilita al objeto para que sea “visitado” por el objeto visitor. Cada objeto de una colección debe implementar esta abstracción para poder ser capaz de ser visitado.

Participantes (2)

- ConcreteElement. Estas clases implementan la interfaz o clase Element y definen la función accept. El objeto de tipo visitor es pasado a este objeto como parámetro de la función accept.
- EstructuraDeObjeto. Es una clase que contiene todos los objetos que pueden ser visitados. Ofrece un mecanismo para iterar los elementos. Puede ser un objeto compuesto.

Colaboraciones

- Los patrones de diseño Iterator y Visitor pueden aplicarse al patrón de diseño Composite.
- El patron de diseño Iterator puede modificar(traverse)? A un Composite. El patrón vistor puede aplicar una operación sobre un Composite.