

Паттерны проектирования

Лекция 4

Итератор

- Сложная структура
- Обход всех ее элементов
- Стандартный подход: использование индексов

```
For(int i=0; i< structure.length(); i++) {  
    doSomething(structure.get(i))  
}
```

Описание

Классификация

Итератор – паттерн поведения объектов.

Описание

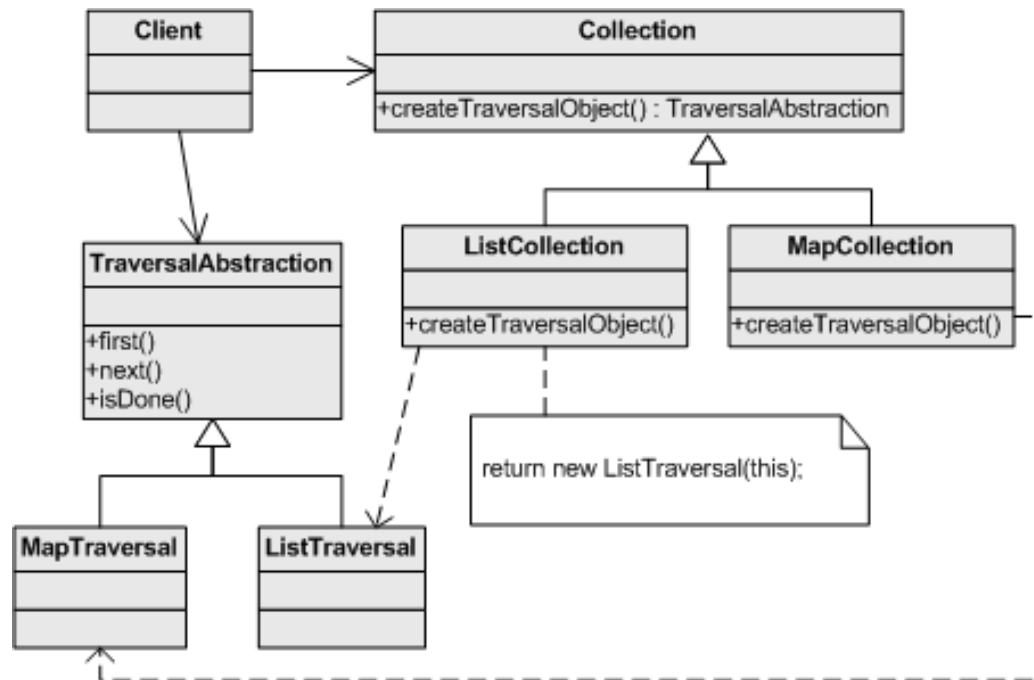
- Последовательный доступ ко всем элементам объекта
- Без раскрытия его внутреннего представления.

Идея

- За обход отвечает отдельный объект
- Хранит состояние обхода



Структура



Результаты

- Единообразный интерфейс обхода различных структур
- Легко добавить новый алгоритм обхода
- Несколько параллельных обходов

Комментарии

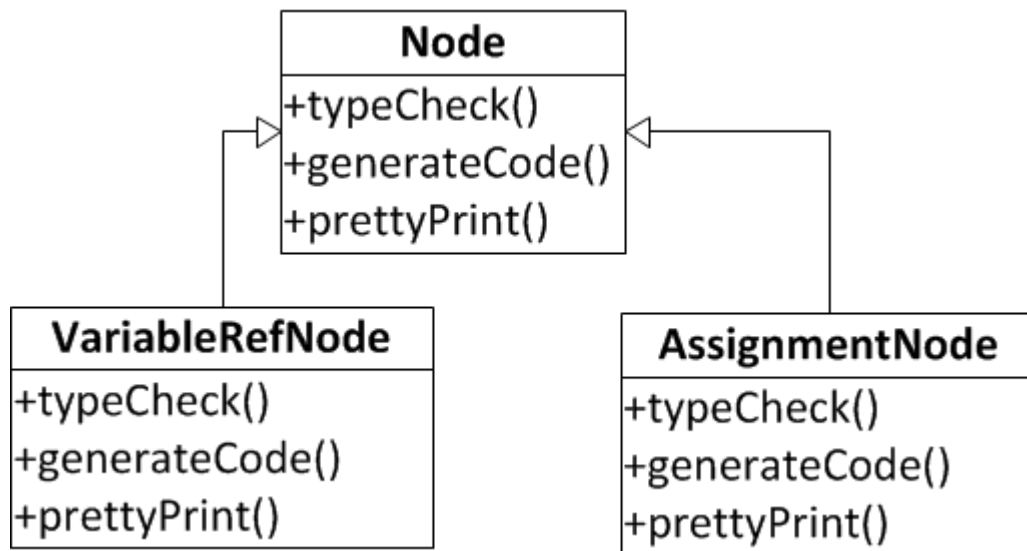
- Внутренние и внешние итераторы

```
public interface InternalIterable<T> {  
    void iterate(Function<T> closure);  
}  
  
public interface Function<T> { void invoke(T t); }
```

- Устойчивые итераторы
- Набор операций итератора
- Итератор достаточно привязан к структуре

Посетитель (Visitor)

- Пример: компилятор и абстрактное синтаксическое дерево
- Множество типов узлов и операций над ними



Описание

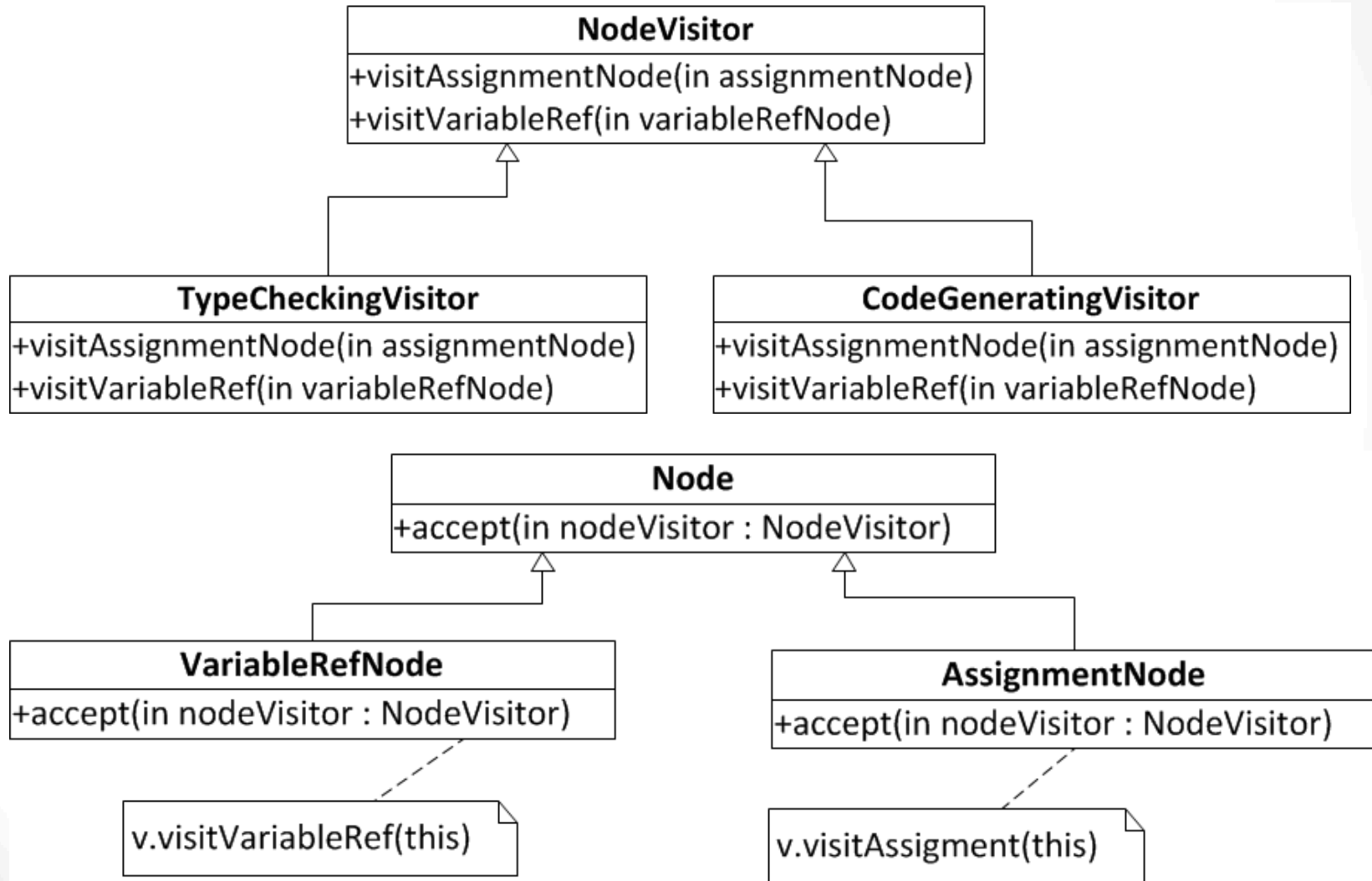
Классификация

Посетитель – паттерн поведения объектов.

Описание

- Описывает операцию, выполняемую с объектом из структуры
- Определение новой операции не затрагивает объекты структуры

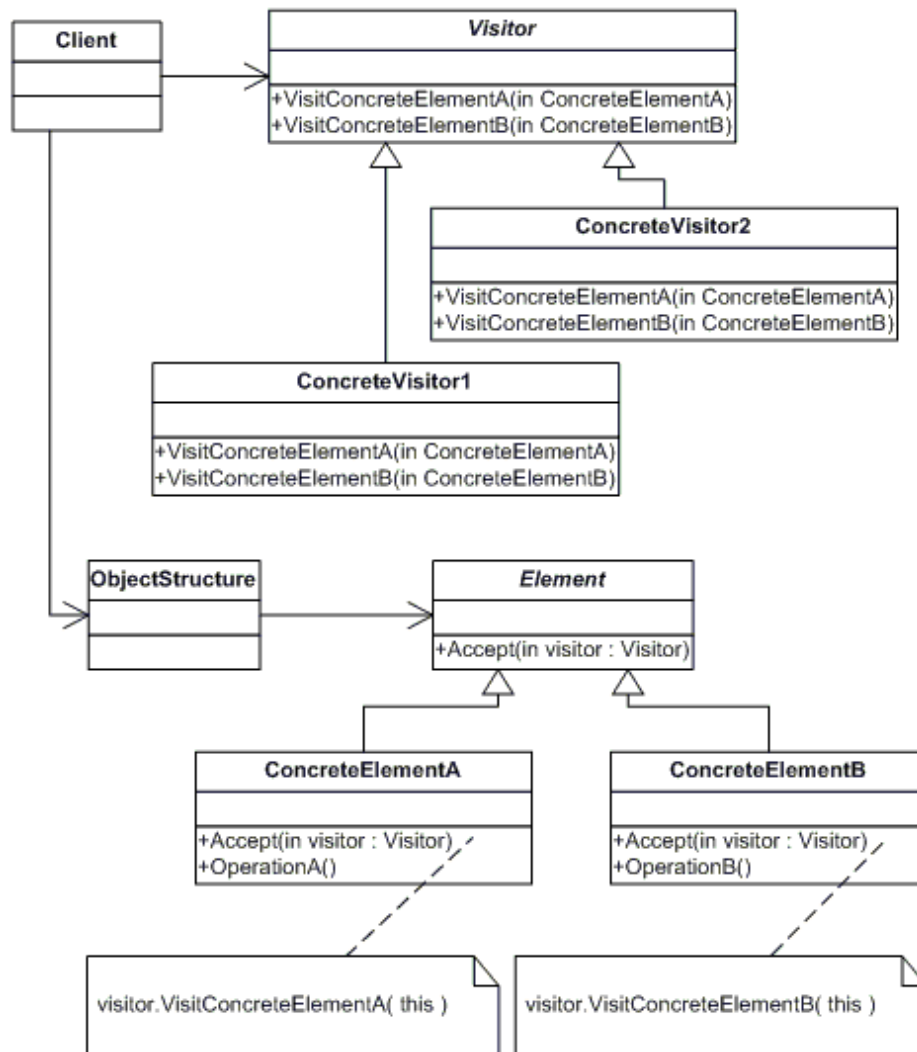
Идея



Применимость

- Для структуры с множеством классов нужно выполнять множество различных операций
- Операции не связаны между собой
- Новые типы элементов появляются редко, а операции часто

Структура



Комментарии

- Добавлять новые операции просто
- Добавлять новые типы объектов сложно
- Может хранить состояние
- Может действовать вне иерархии классов

ССЫЛКИ

- Диаграммы и примеры кода: <http://www.oodeesign.com> и <http://sourcemaking.com>