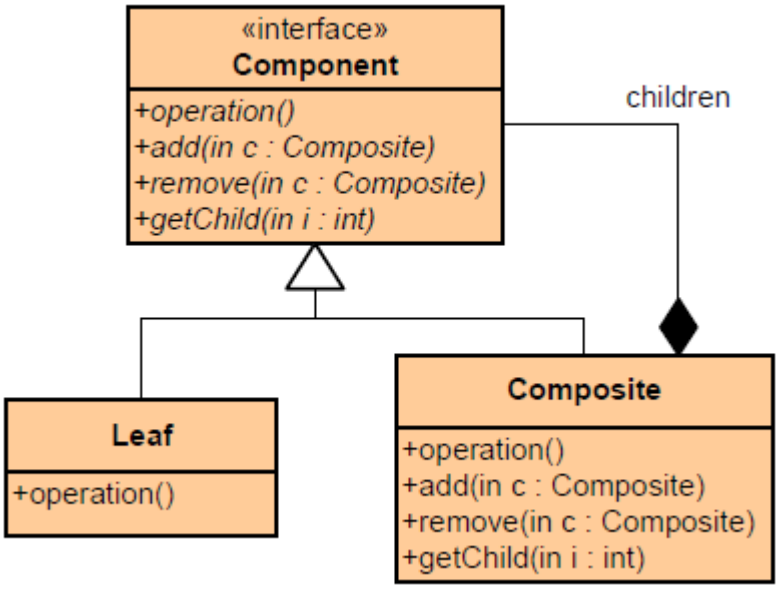
Composite 패턴

객체들의 관계를 트리 구조로 구성하여 객체 하나만 선택할 수도 있고 여러 개를 선택할 수도 있도록 함

타입

Structural Pattern

클래스 다이어그램



예제

composite.cpp

```
#include <iostream>
#include <vector>
#include <string>

using std::cout;
using std::vector;
using std::string;

class Component
{
  public:
   virtual void list() const = 0;
   virtual ~Component(){};
```

```
};

class Leaf : public Component
{
  public:
   explicit Leaf(int val) : value_(val)
   {
   }
   void list() const
   {
      cout << "    " << value_ << "\n";
   }
  private:
   int value_;
};

class Composite : public Component
{
  public:
   explicit Composite(string id) : id_(id)
   {
   }
   void add(Component *obj)
   {
      table_.push_back(obj);
   }
   void list() const
   {
      cout << id_ << ":" << "\n";
      for (vector<Component*>::const_iterator it = table_.begin();
       it != table_.end(); ++it)
      {
         (*it)->list();
      }
   }
  private:
   vector <Component*> table_;
   string id_;
};

int main()
{
   Leaf num0(0);
   Leaf num1(1);
   Leaf num2(2);
   Leaf num3(3);
   Leaf num4(4);
   Composite container1("Container 1");
   Composite container2("Container 2");
```

```
    container1.add(&num0);
    container1.add(&num1);

    container2.add(&num2);
    container2.add(&num3);
    container2.add(&num4);

    container1.add(&container2);
    container1.list();
    return 0;
}
```

참고

위키피디아 Composite 패턴