Decorator Pattern.

6조 발표자 :이재후
조원: 권다솔,박병제,손동현.

Decorator Pattern 목차.

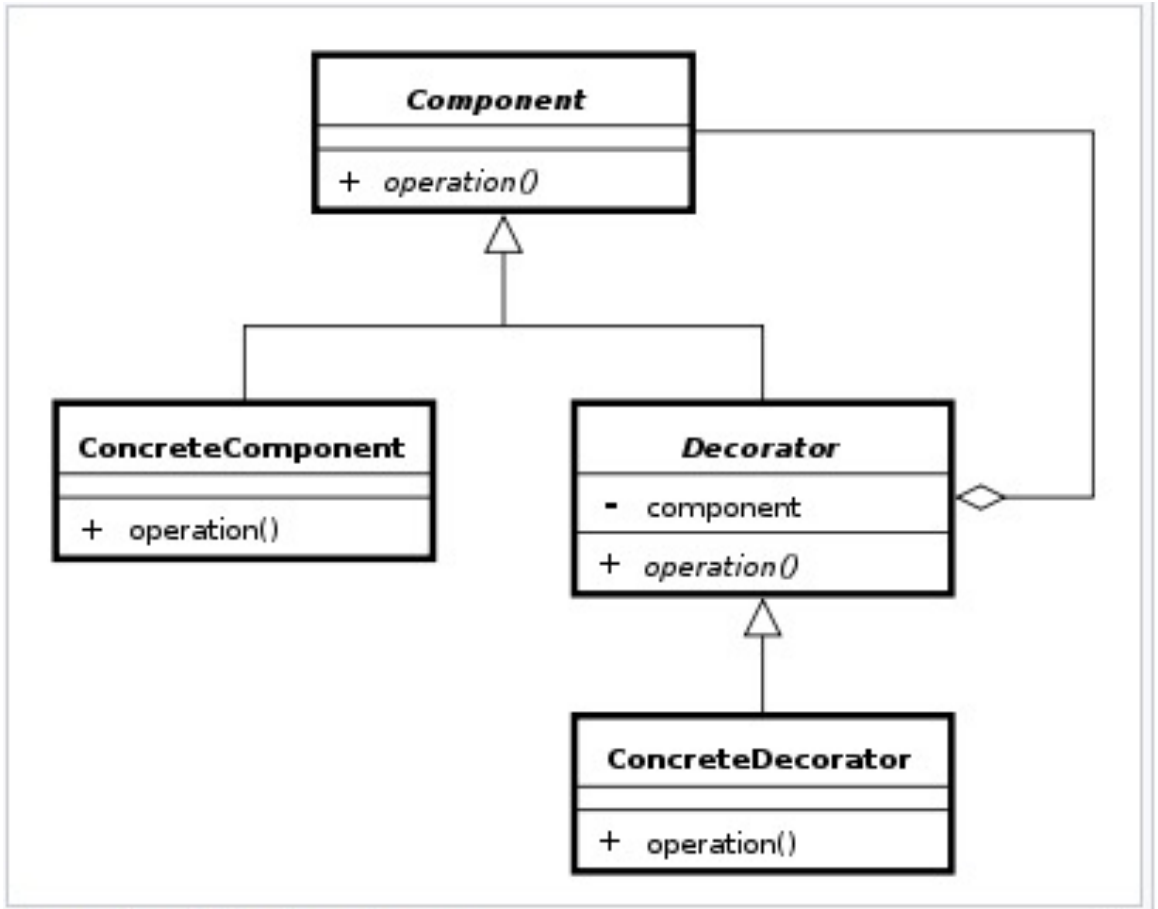
Part 1

정 의

Part 1,

정 의

1. Component
2. Concrete Componet
3. Decorator
4. Concrete Decorator



Decorator UML class diagram

Part 2, 기본 요소

Component

- 실질적인 인스턴스를 컨트롤하는 역할

Concrete Componet

- 기능 추가를 받을 기본 객체

Decorator

- 기능 추가를 할 객체를 위한 추상 클래스

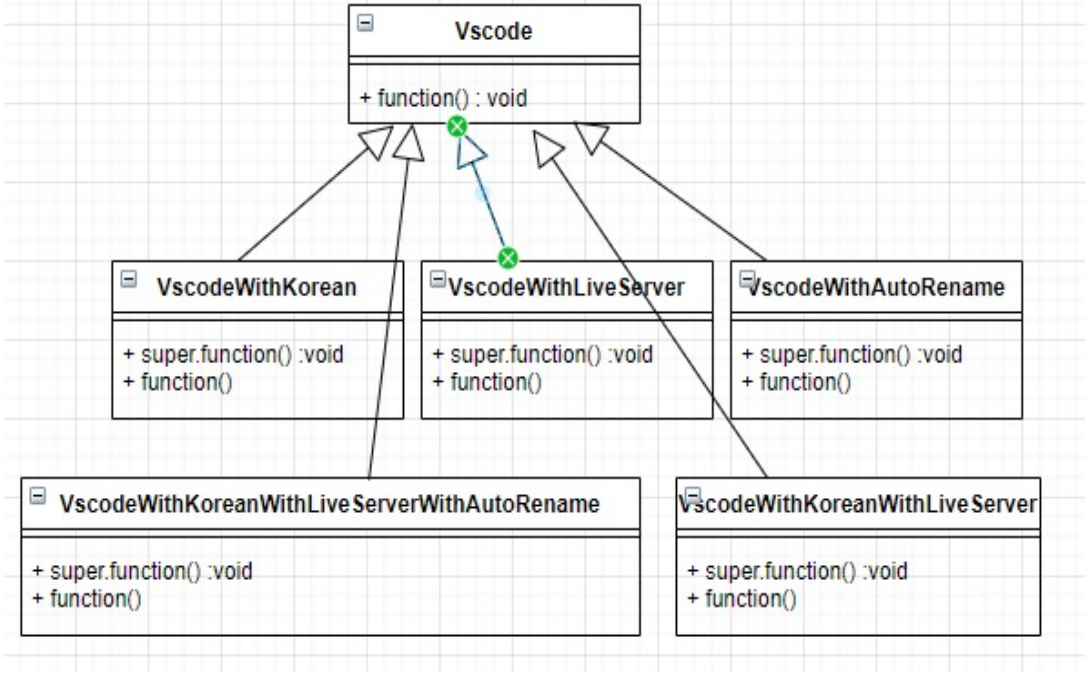
Concrete Decorator

- 실질적인 장식 인스턴스 및 정의이며 추가된 책임의 주체

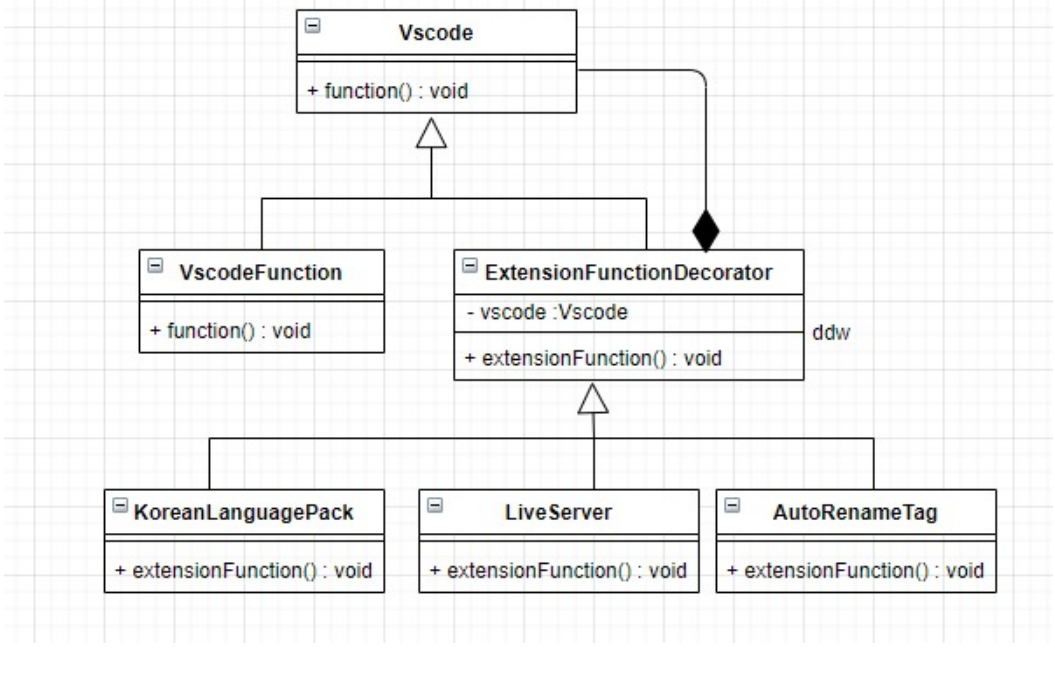
Part 2

예 제

Part 1, Dacorator pattern 관계도



-Dcorator pattern 미적용 시 클라이언트의 새로운 요구마다 객체를 생성을 해야 합니다.



- Dcorator pattern 적용 시 기존 요구 조건은 은 그대로 유지하고 새로운 요구 조건만 늘리는 방식입니다.

Part 3, 소스코드 예제

기본 기능

-Vscode의 기본 기능

추가 기능 (확장 프로그램) – decorate

- Korean Language Pack (한국어 번역) 기능

- Auto Rename Tag (태그 이름 자동 변경) 기능

- Live Server (서버 역할 수행) 기능

Part 1,

Dacorator 미적용 소스

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

```
public class Vscode {
    public void function(){
        System.out.println("vscode 기본 기능");
    }
}
```

- 기본 기능의 Vscode.

- 기능 구현 부분은 콘솔 출력으로 대체.

```
public class VscodeWithKorean extends Vscode{
    public void function(){
        super.function();
        System.out.println("한국어 기능");
    }
}
```

- 한국어 확장 팩이 추가된 VsCode.

(기본 기능 Vscode에 새로운 기능을 추가함.)

Part 1,

Dacorator 미적용 소스

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

- Live Server 기능이 추가된 Vscode

```
public class VscodeWithLiveServer extends Vscode{
    public void function(){
        super.function();
        System.out.println("Live Server 기능 추가");
    }
}
```

- 태그 명 자동 변경 기능이 추가된 Vscode

```
public class VscodeWithAutoRename extends Vscode{
    public void function(){
        super.function();
        System.out.println("태그명 자동 변경 기능");
    }
}
```

- Live Server, 태그명 자동 변경 기능이 추가된 Vscode

```
public class VscodeWithAutoRenameWithLiveServer extends Vscode{
    public void function(){
        super.function();
        System.out.println("태그명 자동 변경 기능");
        System.out.println("Live Server 기능");
    }
}
```

문제점

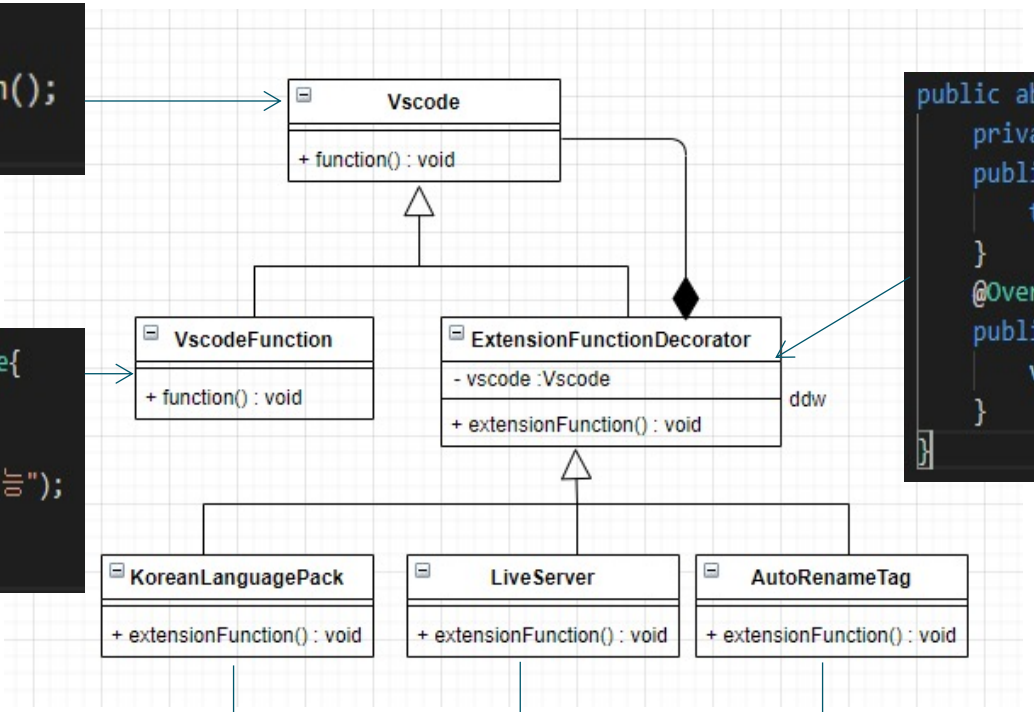
-같은 기능이지만 기존의 기능을 활용 못하고 새로운 객체를 다시 만들어야 합니다.

Part 1, Dacorator 적용 소스

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

```java
public abstract class Vscode {
    public abstract void function();
}
```

- **Component (Vscode)**
  -클라이언트가 사용하는 영역




```java
public class VscodeFunction extends Vscode{
    @Override
    public void function(){
        System.out.println("Vscode 기본 기능");
    }
}
```

- **ConcreteComponent (VscodeFunction)**
  - 기본 기능을 구현.

```java
public abstract class ExtensionFunctionDecorator extends Vscode{
    private Vscode vscode;
    public ExtensionFunctionDecorator(Vscode vscode){
        this.vscode = vscode;
    }
    @Override
    public void function(){
        vscode.function();
    }
}
```

- **Decorator (ExtensionFunctionDecorator)**
  - **추가 기능을 구성하는 요소와 같은 인터페이스 혹은 추상 클래스**

```java
public class KoreanLanguagePack extends ExtensionFunctionDecorator{
    public KoreanLanguagePack(Vscode vscode){
        super(vscode);
    }
    @Override
    public void function(){
        super.function();
        System.out.println("한국어 기능");
    }
}
```

- **ConcreteDecorator**
  - **기본 기능에 추가 되는 추가 기능을 구현**

Part 1,

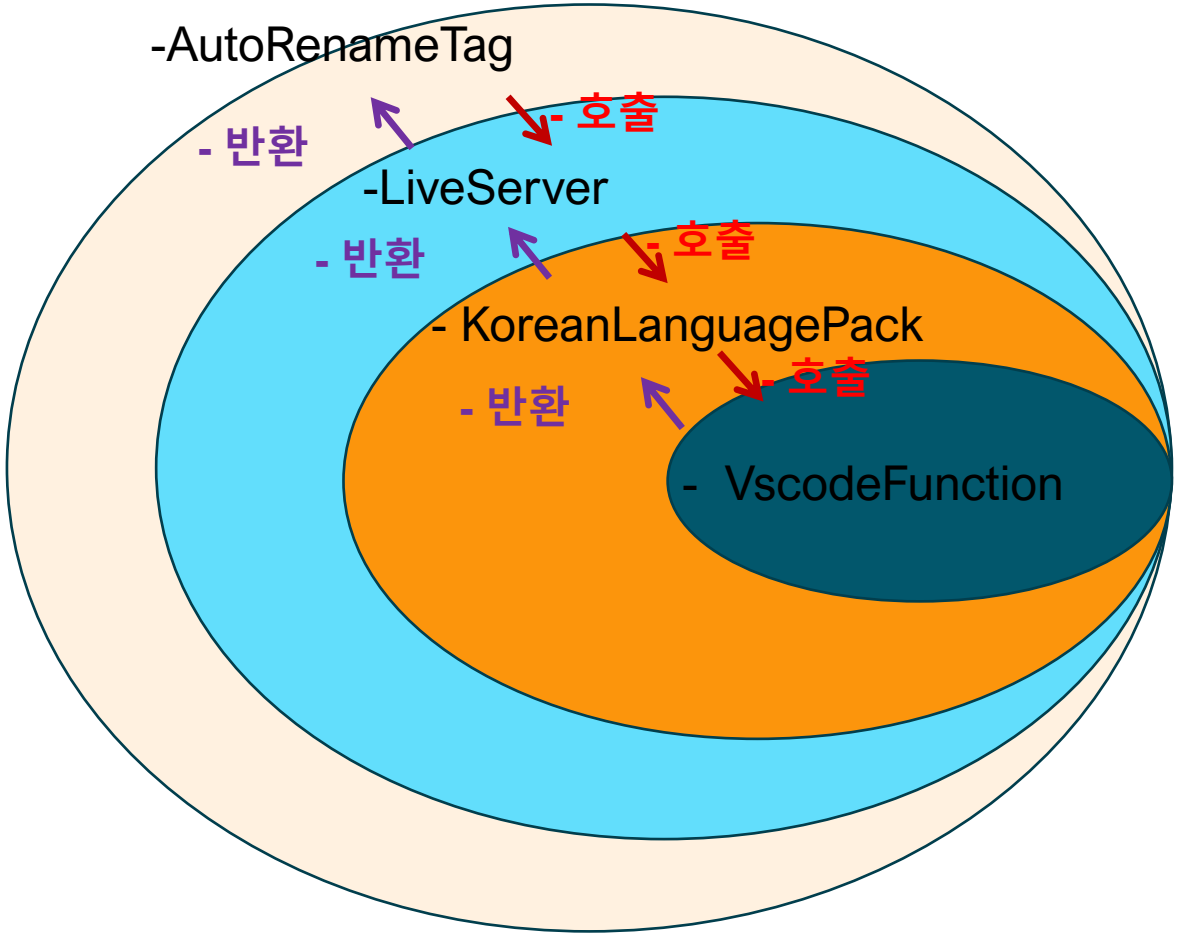
Dacorator 적용 소스

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

- Decorator Pattern 방식

```
public class Main {
    Run | Debug
    public static void main(String args[]){
        Vscode myVscode = new AutoRenameTag(
                                new LiveServer(
                                    new KoreanLanguagePack(
                                        new VscodeFunction())));

        myVscode.function();
    }
}
```



Part 3

장 단 점

Part 3,

제목을 입력하세요

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

장점

OCP(Open-Closed Principle) 디자인 원칙

- 클래스의 확장에 대해서는 열려 있어야 하고 코드 변경에 대해서는 닫혀 있어야 한다.
- 새로운 기능의 추가에 유연하다

단점

- 큰 프로젝트를 진행하면 클래스가 너무 많아진다.

-> 팩토리패턴, 빌더패턴 등 다른 패턴을 활용해 코드를 작성 해야함.

- 정의된 메소드의 개수가 증가하면 복잡하다.
- 객체의 정체를 알기 힘들다.


감사합니다