Short Report Assignment

for Object-Oriented Design Pattern

21700034 YeongHye Kwak

* Analyze and understand the Yuki’s example

1. How the code is modularized

텍스트, 지도, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<Yuki example(Chain of Responsibility) UML Diagram>

스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명

<Main Class(send request(Trouble(i)) to object servers>

As you can see in the screenshot, if the request of Main Class(Trouble(i)) is assigned to the first object server(alice) and that server can’t handle the request, it passes the request to the next object server(bob) connected by the setNext function to handle the request.

This process is repeated in order of alice, bob, charile, diana, elmo, and fred, which results in a design pattern for performing requests on the object server that can handle the request.

1. How the servers’ responsibilities are chained

스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명

<Main Class(send request(Trouble(i)) to object servers>

스크린샷이(가) 표시된 사진

자동 생성된 설명

<Support Class(Repeatedly reused for object servers)>

Could check the servers’ responsibilities are chanined in order of alice, bob, charile, diana, elmo, and fred, because of the setNext function generated in the Main Class.

1. And other your understanding

Since NoSupport, LimitSupport, SpecialSupport, and OddSupport are child class relationships inherited from Support Class, I found that some of the functions in Support class can be modified by using overriding.