

SWIFT Proxy

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Proxy

Structure

Implementation

References

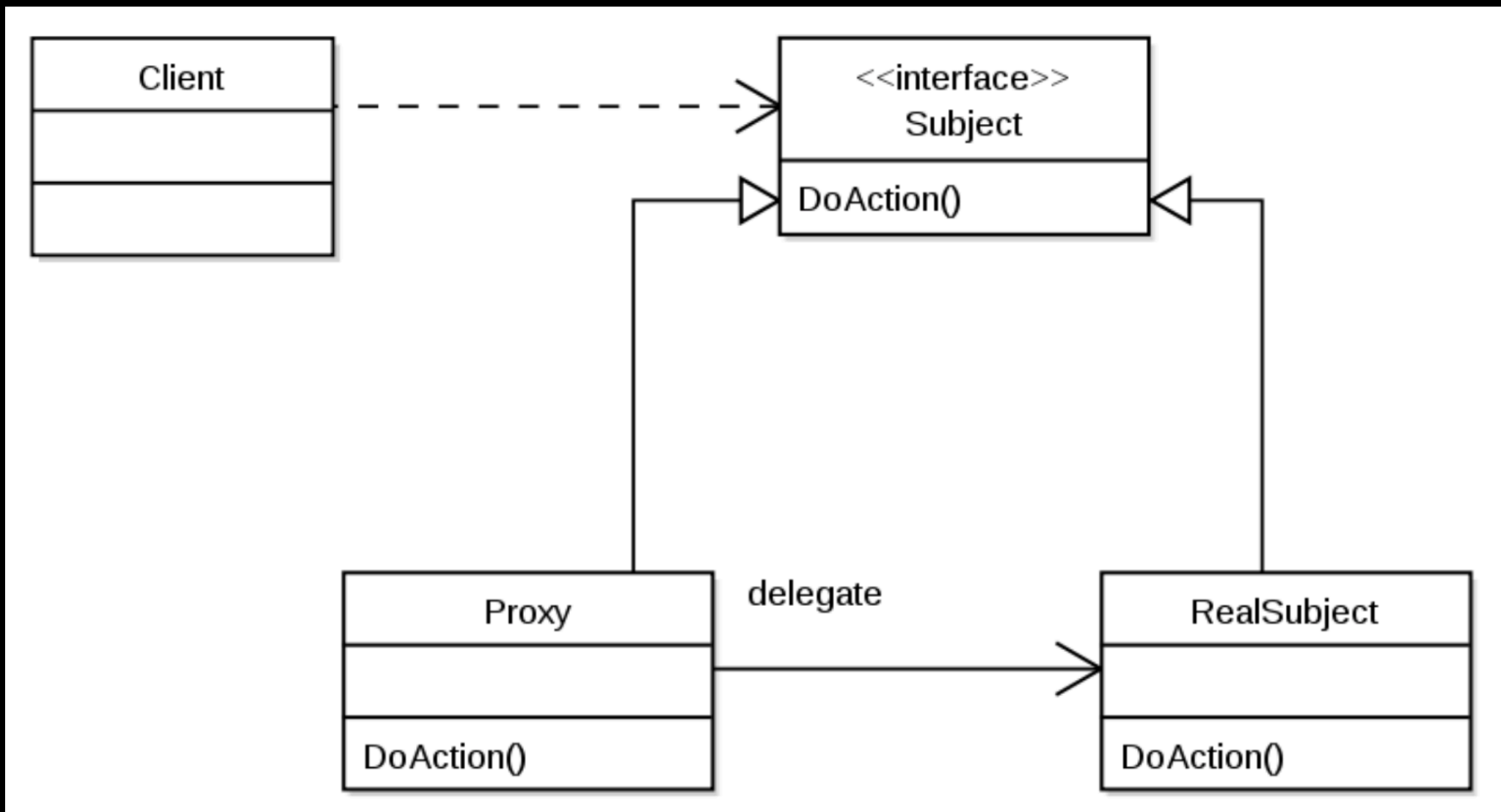
Proxy

Proxy(프록시)패턴은 실제 기능을 수행하는 객체(Real Object) 대신 가상의 객체(Proxy Object)를 사용해 로직의 흐름을 제어하는 디자인 패턴입니다.

즉 Proxy 객체를 이용하여 원래 객체의 변경없이 기능을 특정 방식을 제어할 수 있게 도와줍니다.

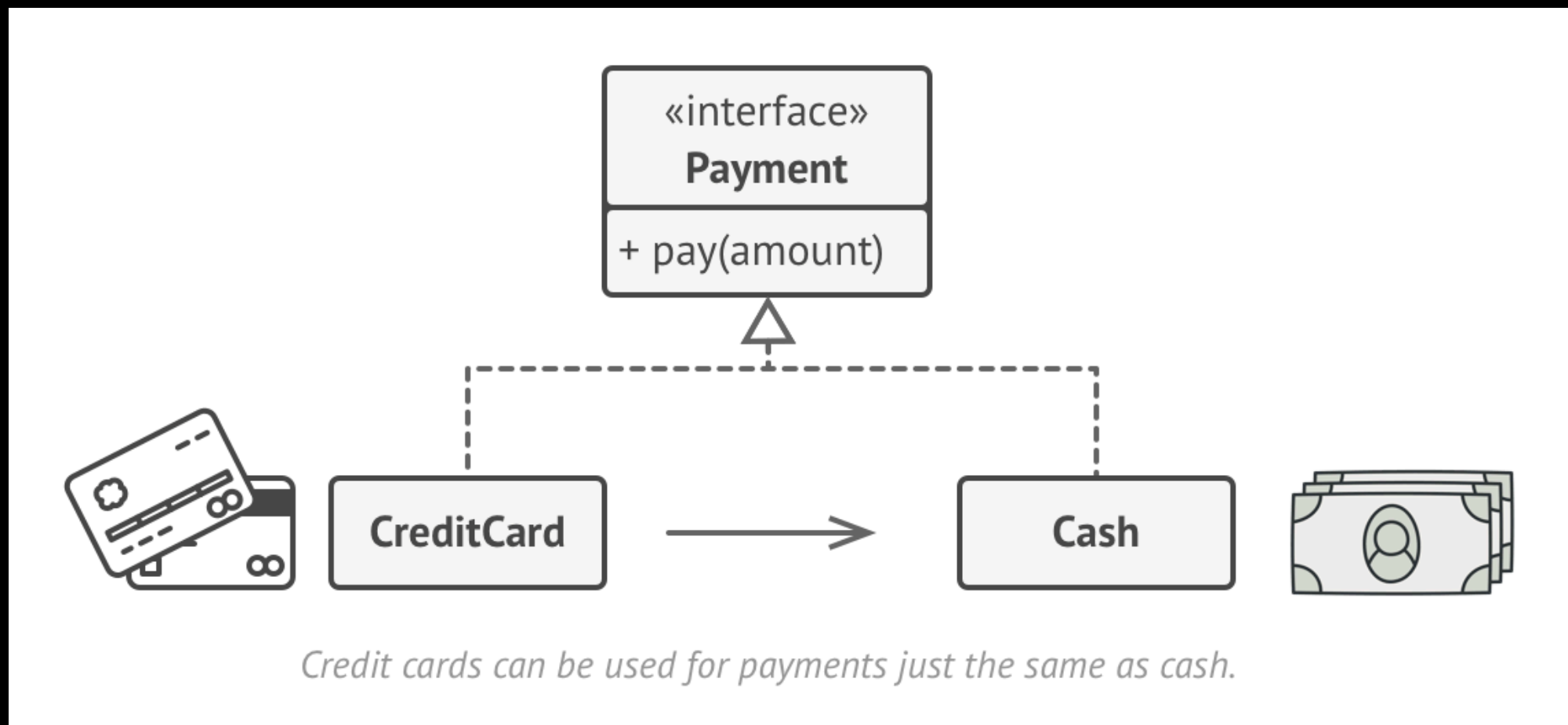
Structure

Proxy 패턴을 UML로 도식화하면 아래와 같습니다.



Structure

Proxy 패턴을 실생활에서의 예시로 보면 아래와 같은 현상을 볼 수 있습니다.



Structure

Subject : Client가 직접 접근하여 기능을 수행하는 추상 부모 클래스, 해당 클래스를 **RealSubject**가 상속을 받아서 실제 기능에 대한 구현을 수행한다.

RealSubject : 실제 기능을 수행하는 객체

Proxy : **RealSubject** 객체 대신에 기능을 위임받아 수행하는 역할을 하는 객체, 해당 **Proxy** 객체가 실제 객체의 기능 및 추가적인 기능을 수행한다.

Implementation

구체적인 구현에 대해서 소스 코드를 통하여 살펴봅니다.

```
protocol Subject {  
    func request()  
}  
  
class RealSubject : Subject {  
    func request() {  
        print("RealSubject request.")  
    }  
}
```

Implementation

```
class Proxy : Subject {  
    private var realSubject: RealSubject  
  
    init(_ realSubject: RealSubject) {  
        self.realSubject = realSubject  
    }  
  
    func request() {  
        print("Proxy request.")  
  
        otherFunction()  
    }  
  
    private func otherFunction() { }  
}
```


Implementation

```
func clientProxy(subject: Subject) {  
    subject.request()  
}
```

```
let realSubject = RealSubject()  
clientProxy(subject: realSubject)  
// RealSubject request.
```

```
let proxy = Proxy(realSubject)  
clientProxy(subject: proxy)  
// Proxy request.  
// RealSubject request.
```

References

[1] Proxy in Swift : <https://refactoring.guru/design-patterns/proxy/swift/example>

[2] Swift World: Design Patterns — Proxy : <https://medium.com/swiftworld/swift-world-design-patterns-proxy-6e2239532a88>

[3] 프록시 패턴 (Proxy Pattern in Swift) : <https://jerome.kr/entry/proxy-pattern>

[4] 프록시 패턴(Proxy Pattern) : <https://jdm.kr/blog/235>

[5] A Design Pattern Story in Swift - Chapter 12: Proxy : <http://audreyli.me/2015/07/07/a-design-pattern-story-in-swift-chapter-12-proxy/>

References

[6] [Design Pattern] 프록시(Proxy) 패턴 : <https://palpit.tistory.com/196>

[7] 안드로이드에서 Dynamic Proxy 사용하기 : <https://www.charlezz.com/?p=759>

[8] 프록시 패턴 : https://ko.wikipedia.org/wiki/프록시_패턴

[9] Proxy Design Pattern : https://sourcemaking.com/design_patterns/proxy

[10] Proxy design pattern in iOS : <http://devmonologue.com/ios/proxy-design-pattern-ios/>

Thank you!