# BoostCamp iOS

MOGAY

# Content

1. Message Selector
2. Proxy
3. Bundle

# Message Selector

**C++**

컴파일 과정에서 객체 메서드에 대한 함수 포인터를 찾아서 고정된 메모리 주소를 바인드 했다가 호출

**Selector**

```
SEL theSelector = @selector(drawSomething);
```

메시지를 받을 객체의 method 중에서 적합한 method를 고르는 역할

컴파일 시점에 미리 바인드할 필요가 없어짐

런타임 중에 메서드를 찾아 늦게 바인드

# Message Selector

**Target & Action**

Target : 특정한 이벤트를 받을 객체

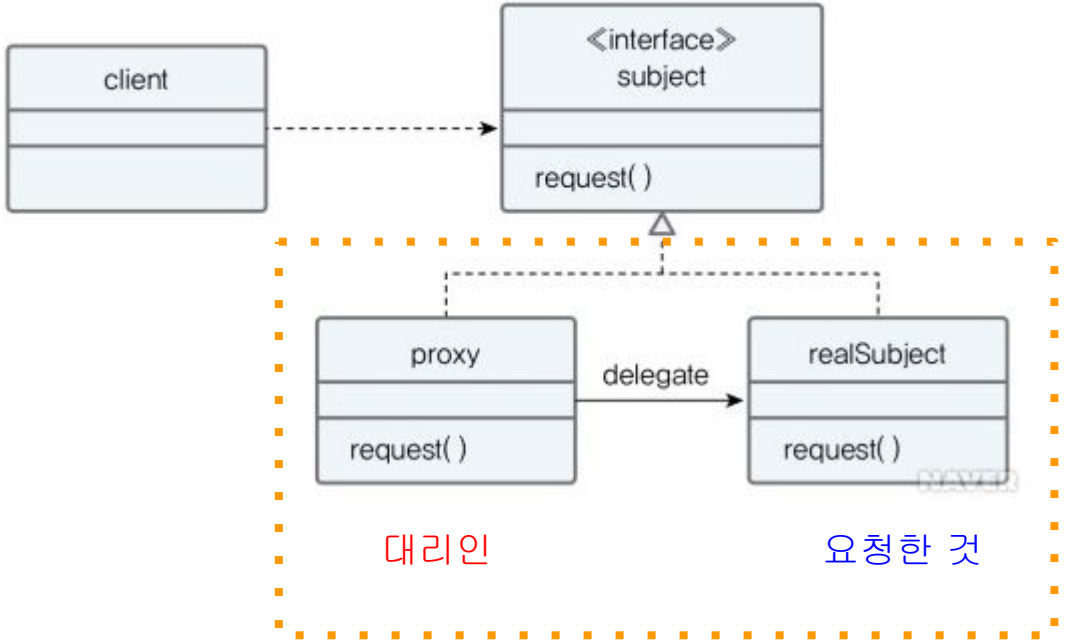
Action : 이벤트를 받아 처리할 method (selector 사용)

**장점**

코드와 분리해서 뷰 객체에 대한 재사용성을 높일 수 있음

selector는 실행 중에 문자열로 지정하는 방식도 가능하기 때문에 특정 객체의 이벤트 처리를 실행 중에 target과 action으로 지정해서 연결 가능

# Proxy Pattern




client가 realSubject 클래스를 사용하려고 할 때 subject 인터페이스를 통해서 요청하게 되고 어떤 객체를 통해서 동작하는지는 알 수 없다.

proxy 클래스는 realSubject 객체를 참조하고 있으며 client 요청에 적절한 행동을 취한다.

realSubject는 실제로 동작하는 클래스이다.

# Proxy Pattern 예시

NSProxy 클래스

➥ 숨겨진 최상위 객체

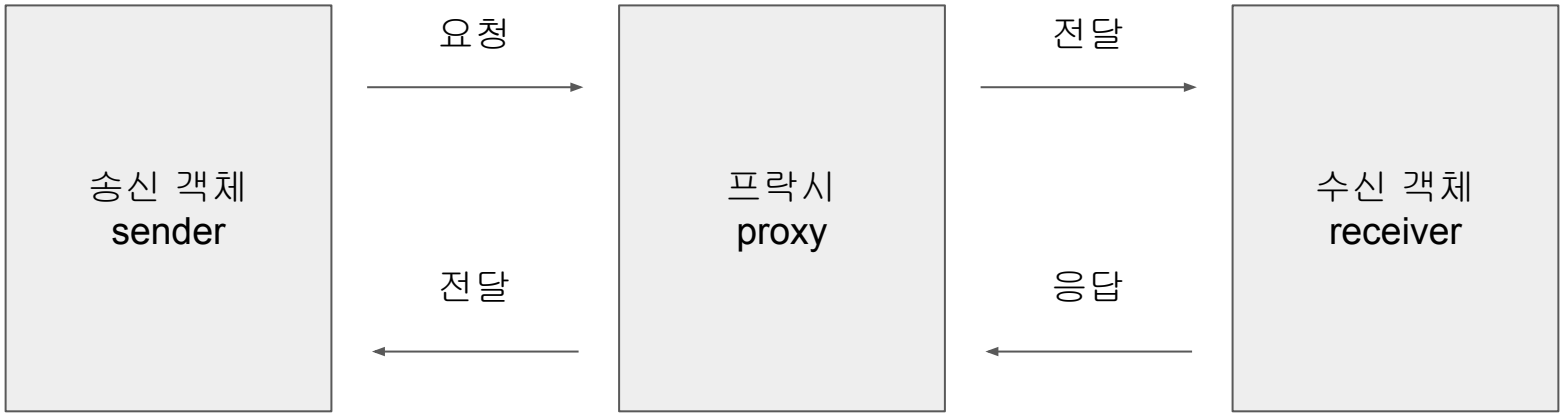
➥ 코코아에서 유일하게 NSObject에서 상속받지 않고 **NSObjectProtocol만 구현**

➥ 단지 다른 객체를 대신하는 동작을 위해 필요한 **인터페이스만 선언**한 추상화 클래스

➥ NSProxy 인터페이스를 상속받아 만든 프락시 객체는 **원하는 메시지를 다른 객체로 전달**하는 역할을 한다.

➥ 구현되어 있는 proxy 클래스: NSDistantObject

# Proxy 객체 동작 방식



# Bundle???

**번들**(bundle) - 정의된 파일 확장자를 갖는 디렉터리로, 개념상 관련된 파일들을 모두 하나로 묶어주는 역할.

보통 번들은 애플리케이션, 프레임워크나 플러그인의 실행 파일을 포함하고 있다. 이런 번들들은 실행 코드란 걸 알려주는 파일이 있으며, 그리고 인터페이스 빌더나 스트링 같은 같은 리소스 파일, 틀, 사진, 음악 등등을 모두 다 집어넣고 있다.

# Bundle Pattern

Bundle 구조를 사용해서 프로그램, 프레임 워크 내의 복잡성을 감추는 디자인 패턴

Info.plist - application의 기본정보를 key - value 형식으로 저장, key 값은 Bundle 클래스에 선언되어 있다.

리소스 지역화

- 리소스 지역화 디렉터리는 {언어}_{지역}.lproj 형식으로 만들어 각 지역에 대응하는 리소스를 저장.

# iOS 앱 번들

iOS 앱은 실행 파일과 리소스 전체가 하나의 디렉터리로 구성

IPA 파일 - 앱 스토어에서 배포하는 앱 파일 ( 인증서 구매 정보 저장)

스토리보드, 인터페이스 파일을 컴파일하고 이미지도 데이터 인코딩을 변경해서 저장

번들 구조는 앱, 프레임워크, 플러그인등 다양한 곳에서 활용!

# 감사합니다