

SWIFT Singleton

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Singleton

Class constant

Nested struct

References

Singleton

Singleton(싱글톤) 패턴은 객체 생성과 관련한 디자인 패턴으로서 객체 생성을 단 한번만 생성하여 어디서든 참조하여 사용할 수 있도록 도와주는 디자인 패턴입니다.

싱글톤은 기존 언어들에서도 많이 사용되고 있는 패턴으로서 적절하게 사용하게 되면 좋은 방법이 될 수 있으나 너무 많은 참조의 형태로 커지게 되면 외부에 강한 의존성으로 인하여 안 좋은 설계가 될 수 있습니다.

Singleton

Swift에서의 싱글톤 생성 방식은 크게 보면 아래의 세 가지 방식으로 사용가능합니다.

- 1) 클래스 상수(Class constant)
- 2) 중첩 구조체(Nested struct)
- 3) `dispatch_once`(Swift 3.0 부터는 사용불가)

본 강의에서는 위의 2가지 방식에 대해서 하나씩 코드를 통하여 살펴보겠습니다.

Class constant

- Class 내부에서 sharedInstance 로 자체 객체 생성 및 반환
- Private init() 함수 선언으로 외부 생성하지 못하도록 제한

```
class SingletonA {  
    static let sharedInstance = SingletonA()  
  
    private init() {  
        print("SingletonA init");  
    }  
  
    func test() {  
        print("SingletonA test");  
    }  
}
```

```
SingletonA.sharedInstance.test()  
SingletonA.sharedInstance.test()
```

```
// Print SingletonA init  
// Print SingletonA test  
// Print SingletonA test
```

```
var a:SingletonA = SingletonA() // 컴파일 에러, 외부에서 생성 및 초기화 불가
```

Nested struct

- Class 내부의 중첩 구조체에서 정적 상수 사용으로 객체 생성 및 반환
- Private init() 함수 선언으로 외부 생성하지 못하도록 제한

```
class SingletonB {  
    class var sharedInstance: SingletonB {  
        struct Static {  
            static let instance: SingletonB = SingletonB()  
        }  
  
        return Static.instance  
    }  
  
    private init() {  
        print("SingletonB init");  
    }  
  
    func test() {  
        print("SingletonB test");  
    }  
}
```

```
SingletonB.sharedInstance.test()  
SingletonB.sharedInstance.test()
```

```
// Print SingletonB init  
// Print SingletonB test  
// Print SingletonB test
```

```
var b:SingletonB = SingletonB() // 컴파일 에러, 외부에서 생성 및 초기화 불가
```

References

[1] 싱글톤 패턴 : <https://jerome.kr/entry/singleton-pattern?category=1114713>

[2] [Swift-Design Pattern] 싱글톤 패턴 (Singleton Pattern) : <http://throughkim.kr/2019/09/04/swift-singleton/>

[3] [NAROTi][iOS 개발] Singleton Pattern : <https://velog.io/@naroti/iOS-개발-Singleton-Pattern-q4k3uzgf0n>

[4] Singleton Pattern(swift) : <https://linsaeng.tistory.com/9>

[5] [Swift]Singleton 패턴 사용 방법 3가지 : <http://minsone.github.io/mac/ios/singleton-in-swift>

References

- [6] Design Pattern - Singleton : <https://ehdrjsdlzzzz.github.io/2018/10/28/Design-Pattern-Singleton/>
- [7] [SWIFT] 싱글톤(Singleton) 패턴 : <https://faith-developer.tistory.com/68>
- [8] [Swift 개념] Singleton Design Pattern : <https://etst.tistory.com/93>
- [9] 싱글톤 패턴(Singleton pattern)을 쓰는 이유와 문제점 : <https://jeong-pro.tistory.com/86>
- [10] Swift 에서 싱글톤(Singleton) 패턴 사용하기 : <https://comxp.tistory.com/331>

Thank you!