SWIFT
Class & Structure

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Class & Structure

구조체와 클래스는 OOP(Object-Oriented Programming)를 위한 필수 요소로 프로그램의 코드를 추상화하기 위해 사용하는 개념으로서 변수와 함수 등을 하나의 집합체로 구성하고자 할 경우 사용하는 데이터 타입

공통점

- 프로퍼티 정의가 가능
- 메소드 정의가 가능
- 초기화(initializer) 정의가 가능
- 확장(extension) 사용 가능
- 프로토콜(protocol) 사용 가능
- 서브스크립트(subscript) 사용이 가능

Class & Structure

차이점

- 구조체는 값 타입(Value type)
- 클래스는 참조 타입(Reference type)
- 구조체는 상속이 불가능
- 타입캐스팅은 클래스의 인스턴스에만 허용
- 디이녈라이저는 클래스의 인스턴스에만 활용
- 구조체에서는 AnyObject로 타입캐스팅이 불가능
- 구조체는 생성자를 구현하지 않을 시 기본 initializer를 사용 가능
- 클래스는 Reference counting으로 메모리 관리가 가능

Instance & Initialize

구조체

```
struct Person {
    var name: String
    var age: Int
}

// 프로퍼티 이름(name, age)으로 자동 생성된 이니셜라이저를 사용하여 구조체를 생성합니다.
var bill: Person = Person(name: "Bill", age: 99)
bill.age = 30
bill.name = "Bill"

var joyce = bill;
joyce.age = 50;

print(bill.age) // 30
print(joyce.age) // 20, joyce는 별도의 주소로 생성
```

Instance & Initialize

클래스

```
class Person {
	var name: String = "" // 값을 초기화하지 않을 경우는 기본 초기화 함수를 생성하여야 한다.
	var age: Int = 0 // 값을 초기화하지 않을 경우는 기본 초기화 함수를 생성하여야 한다.

	// 기본 초기화 함수인 init 함수이며 변수 선언과 동시에 값을 초기화하지 않는 경우
	// 반드시 선언하여 변수 값을 설정해주어야 한다.
	init(name: String, age:Int)
	{
		self.name = name
		self.age = age
	}
}

var bill: Person = Person()
bill.age = 30
bill.name = "Bill"

print(bill.age)

var joyce = bill
joyce.age = 20;

print(bill.age) // 20, joyce 객체가 bill의 인스턴스 주소와 동일하여 bill의 나이도 같이 변경됨
print(joyce.age) // 20
```

Deinitialization

클래스

클래스의 경우 메모리 해제시에는 deinit 메서드에 처리할 코드를 넣어서 별도의 처리를 할 수 있습니다.

```
class Person {
        var name: String
        var age: Int

        init(name: String, age:Int)
        {
            self.name = name
            self.age = age
        }

        deinit {
              print("Person 클래스의 인스턴스가 소멸됩니다.")
        }
}
```

When choose Structures?

애플은 가이드라인에서 다음 조건 중 하나 이상에 해당된다면 구조체를 사용하기를 권장

1) 연관된 간단한 값의 집합을 캡슐화 하는 것만이 목적일 때

2) 캡슐화된 값이 참조되는 것보다 복사되는 것이 합당할 때

3) 저장된 프로퍼티가 참고보다는 복사되기를 기대하는 경우

4) 프로퍼티나 메소드 등을 상속할 필요가 없는 경우

References

[1] 클래스과 구조체 (Classes and Structures) : https://jusung.gitbook.io/the-swift-language-guide/language-guide/09-classes-and-structures

[2] Swift struct vs. class 차이점 비교 분석 : https://www.letmecompile.com/swift-struct-vs-class-차이점-비교-분석/

[3] Swift - 구조체 클래스 : https://blog.yagom.net/530/

[4] Swift :: 구조체와 클래스 차이 (struct VS class) : https://shark-sea.kr/entry/Swift-구조체와-클래스-차이-struct-vs-class

[5] 오늘의 Swift 상식 (Struct, Class) : https://medium.com/@jgj455/오늘의-swift-상식-struct-class-60fa5fd2218d

References

[6] Swift - 언제 class 대신 struct 를 사용하는가 : http://seorenn.blogspot.com/2016/04/swift-class-struct.html

[7] Swift 구조체와 클래스 : https://hwiveloper.dev/2017/05/31/swift-struct-and-class/

[8] [iOS][Swift] 구조체와 클래스 : Difference between Struct and Class : https://caution-dev.tistory.com/6

[9] [Swift]Class와 Structure 정리 : http://minsone.github.io/mac/ios/swift-classes-and-structures-summary

[10] [Swift] 구조체와 클래스의 차이점 (Mutating) : http://blog.davepang.com/post/353

Thank you!