**CS193p(Lecture5) 2023**

**protocol(Equatable)**

Equatable은 타입 간 비교 연산을 위해 필수적으로 구현해야 하는 프로토콜이다. 일반적인 자료형을 위한 Equatable은 미리 구현이 돼있지만 구조체나 클래스의 비교 연산에 쓰이는 Equatable은 구현되어 있지 않다.

구조체나 열거형(연관값이 없는 경우만 해당)은 Equatable 프로토콜을 채택하는 것만으로도 비교 연산이 가능해진다. 반면에 클래스는 직접 static == 메서드를 구현해서 써야 한다. 구현은 lhs, rhs의 특정 변수를 비교하는 방식으로 한다. 만약 구조체나 열거형에서 모든 변수가 아닌 특정 변수만을 기준으로 동일성을 판단하고 싶으면 직접 == 메소드를 구현해 해당 변수만을 기준으로 삼도록 하면 된다.

ex)

struct Something : Equatable {

var fir = “”

var sec = 2

var third = true

static func == (lhs: Something, rhs: Something) -> Bool {

return lhs.fir == rhs.fir

}

}

let something1 = Something.init(fir: “”, sec: 2, third: true)

let something2 = Something.init(fir: “”, sec: 7, third: false)

something1 == something2 // true

**Don’t Care**

**ForEach Identity, Hashable, Identifiable**

**CustomDebugStringConvertible**

**ViewModel Intent**

**인자로 전달되는 값은 상수이다.**

**Optional은 enum**

값이 없을 경우 nil과 연관된 Optional.none이 되고 값이 있을 경우 Optional.some<T>이 되는 원리이다.