

프로토콜

- 프로토콜은 특정 역할을 수행하기 위한 메서드, 프로퍼티, 기타 요구사항 등의 청사진을 정의한다.
- 구조체, 클래스, 열거형은 프로토콜을 채택(Adopted) 해서 특정 기능을 수행하기 위한 프로토콜의 요구사항을 실제로 구현할 수 있다.
- 어떤 프로토콜의 요구사항을 모두 따르는 타입은 그 프로토콜을 준수한다(Conform) 고 표현한다.
- 타입에서 프로토콜의 요구사항을 충족시키려면 프로토콜이 제시하는 청사진의 기능을 모두 구현해야 합니다. 즉, 프로토콜은 기능을 정의하고 제시 할 뿐이지 스스로 기능을 구현하지는 않는다.

```
1 protocol Talkable {
2
3     // 프로퍼티 요구
4     var topic: String { get set }
5     var language: String { get }
6
7     // 메서드 요구
8     func talk()
9
10    // 이니셜라이저 요구
11    init(topic: String, language: String)
12 }
```

요구

저장프로퍼티는 반드시 var키워드를 사용

get은 읽기만 가능해도 상관 없다는 뜻이며 get과 set을 모두 명시하면 읽기 쓰기 모두 가능한 프로퍼티여야 합니다

```

15 struct Person: Talkable {
16     // 프로퍼티 요구 준수
17     var topic: String
18     let language: String
19
20     // 읽기전용 프로퍼티 요구는 연산 프로퍼티로 대체가 가능합니다
21     // var language: String { return "한국어" }
22
23     // 물론 읽기, 쓰기 프로퍼티도 연산 프로퍼티로 대체할 수 있습니다
24     // var subject: String = ""
25     // var topic: String {
26     //     set {
27     //         self.subject = newValue
28     //     }
29     //     get {
30     //         return self.subject
31     //     }
32     // }
33
34     // 메서드 요구 준수
35     func talk() {
36         print("\(topic)에 대해 \(language)로 말합니다")
37     }
38
39     // 이니셜라이저 요구 준수
40     init(topic: String, language: String) {
41         self.topic = topic
42         self.language = language

```

프로토콜 상속

- 프로토콜은 하나 이상의 프로토콜을 상속받아 기존 프로토콜의 요구사항보다 더 많은 요구사항을 추가할 수 있다.
- 프로토콜 상속 문법은 클래스의 상속 문법과 유사하지만, 프로토콜은 클래스와 다르게 다중상속 가능
- 클래스 상속과 프로토콜 상속을 동시에 채택하려면 클래스를 먼저 명시하고 그 뒤에 채택할 프로토콜 목록을 작성한다.

```

1 protocol 프로토콜 이름: 부모 프로토콜 이름 목록 {
2     /* 정의부 */
3 }

```

```
1 protocol Readable {
2     func read()
3 }
4 protocol Writable {
5     func write()
6 }
7 protocol ReadSpeakable: Readable {
8     func speak()
9 }
10 protocol ReadWriteSpeakable: Readable, Writable {
11     func speak()
12 }
13
14 struct SomeType: ReadWriteSpeakable {
15     func read() {
16         print("Read")
17     }
18     func write() {
19         print("Write")
20     }
21     func speak() {
22         print("Speak")
23     }
24 }
25
```