

프로토콜로 객체 전달하기

프로토콜을 준수하자

객체를 어떻게 전달할 것인가?

- 지금까지 사용한 방법 : 객체 선언 후 직접 전달

```
// 인풋뷰 구조체 선언
let inputView = InputView()
// 체크 기능을 이용하기 위해 체커 선언
let checker = Checker()
// 포인터 선언
let pointerMaker = PointerMaker()
```

- 고민할 필요가 없음

객체를 어떻게 전달할 것인가?

- 문제 : 다른 종류의 객체를 한 함수로 전달할것
- 점, 직선, 삼각형, 사각형 의 객체 존재
- 조건문으로 어떤 객체인지 판별은 가능, 그러나 리턴이 불가능
- 함수의 리턴형을 통일할 수는 없는건가...?

* 방법은 여러가지 : 상속, 프로토콜 등등

선택은 프로토콜

요구조건에서 구조체를 사용하라고 명시 - 클래스 상속 불가

계획 :

- 각 구조체들이 프로토콜을 준수
- 객체를 판별하는 함수가 해당 프로토콜 타입으로 리턴
- 프린트 하는 함수 입력값으로 프로토콜 사용
- 프로토콜의 함수 리턴값을 받아서 프린트

- 프로토콜 정의

```
/// 좌표를 받아서 포인터 객체를 리턴하기 위한 프로토콜 선언
protocol Points{
    // 좌표가 몇개든 마이포인트 배열로 리턴
    func getPoints()->[MyPoint]
    func getMessage()->String
}
```

- 포인트 배열을 리턴하는 함수
- 각 포인트를 프린트하기 위해서 포인트 배열로 리턴
- 메시지를 출력하는 함수
- 요구조건 : 직선 이후부터는 메시지 출력 필요

- 직선 객체 정의

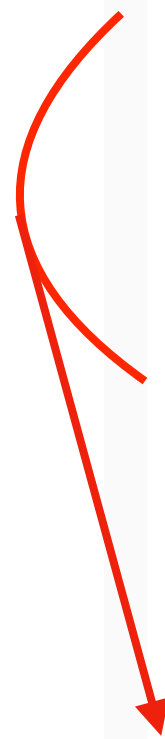
```
struct MyLine : Points {
    private var pointA = MyPoint(x: 0, y: 0)
    private var pointB = MyPoint(x: 0, y: 0)

    /// 마이포인트를 받아서 생성
    init (pointA : MyPoint, pointB : MyPoint){
        self.pointA = pointA
        self.pointB = pointB
    }

    /// 두 점 사이의 거리를 리턴
    func distanceBetweenPonints()->Double {
        let width = self.pointA.getXAxis() - self.pointB.getXAxis()
        let hight = self.pointA.getYAxis() - self.pointB.getYAxis()
        let distance = sqrt(Double((width * width) + (hight * hight)))
        return distance
    }

    /// 프로토콜을 준수
    func getPoints() -> [MyPoint] {
        return [self.pointA, self.pointB]
    }

    func getMessage() -> String {
        return ("두 점 사이의 거리는 \(distanceBetweenPonints())")
    }
}
```



- 출력 객체 정의

```
///프린트용 구조체
```

```
struct OutputView{
```

```
    /// 좌표축을 프린트
```

```
    func drawAxis(){
```

```
        // 화면 지우기
```

```
        print("\(ANSIcode.clear)\(ANSIcode.home)")
```

```
        // 좌표 그리기
```

```
        print("\(ANSIcode.text.redBright)\(ANSIcode.axis.draw())")
```

```
    }
```

```
    /// 입력된 자료를 출력
```

```
    private func drawPoint(myPoints : MyPoint){
```

```
        // 표시용 특수문자
```

```
        let mark = "⦿"
```

```
        // 입력받은 위치로 커서 이동
```

```
        print("\(ANSIcode.cursor.move(row:25-myPoints.getYAxis(), col: myPoints.getXAxis()*2+3))\((mark)")
```

```
        // 특수문자 출력
```

```
        print("\(ANSIcode.cursor.move(row:26, col: 0))")
```

```
    }
```

```
    /// 프로토콜을 받아서 좌표축에 포인터들을 프린트
```

```
    func drawPoints(myPoints:Points){
```

```
        let points = myPoints.getPoints()
```

```
        for point in points {
```

```
            drawPoint(myPoints: point)
```

```
        }
```

```
        print(myPoints.getMessage())
```

```
    }
```

```
}
```

- 프로토콜을 준수하여 객체 전달 성공
- 이 방법은 객체의 추상화이기도 함 👍
 1. 객체가 프로토콜을 준수
 2. 프로토콜 타입으로 리턴
 3. 받는 함수는 프로토콜 타입을 인자로 받음
 4. 어떤 타입의 객체인지 모르는채로 함수 사용

- 추후 과제 : 프로토콜 위임 (Delegation) 패턴 공부
- 다음 코드리뷰 예정 : String.index , 스위프트 코딩 스타일