

과 목 명 : iOS프로그래밍실무

학 과 : 컴퓨터소프트웨어

학 번 : 201822048

     반 : A

이 름 : 황재하

제 출 일 : 2023.05.03

담당교수 : 한성현 교수님

**기말고사 문제**

Open API 기반 iOS앱 개발(2) : 박스오피스 앱

1. 네트워킹(URLSession)
2. App Transport Security(ATS)
3. 1급 객체(first class object), 1급 시민(first class citizen)
4. 클로저(Closure)
5. 후행 클로저(trailing closure)
6. 디폴트 매개변수(아규먼트)
7. failable initializer
8. JSON파싱을 쉽게 하기 위한 구조체 만들기(quicktype)
9. static metatype
10. throwing function과 예외처리(exception handling) : do-try-catch
11. closure 내부에서 self. 사용하는 경우
12. reloadData()와 main thread
13. 비동기 처리 : main 스레드, 백그라운드 스레드
14. 옵셔널 바인딩에 guard-let 활용(if-let보다 좋음)
15. Resolve Auto Layout Issues
16. 앱을 실행하면 어제 날짜로 자동 조회하기

**실습.1**

URL 만들기  
 URL(string: String)

URLSession 만들기  
URLSession(configuration: URLSessionConfiguration)

URLSession 인스턴스에게 task주기

URLSession인스턴스.dataTask(with:completionHandler:)

4. task시작하기

URLSessionTask인스턴스.resume()

**실습.2**

func getData(){ //1

if let url = URL(string: movieURL) { //2

let session = URLSession(configuration: .default)

let task = session.dataTask(with: url) { (data, response, error) in//3

if error != nil { print(error!)

return }

ifletJSONdata=data{ //4

let dataString = String(data: JSONdata, encoding: .utf8) print(dataString!)

let decoder = JSONDecoder()

do{ //5

let decodedData=try decoder.decode(MovieData.self,from:JSONdata) self.movieData = decodedData

DispatchQueue.main.async { self.table.reloadData()

}

}catch{ print(error)

} //5

} //4

} //3

task.resume()

}//2

} //1

func getData(){ //1

guard let url = URL(string: movieURL) else { return }

let session = URLSession(configuration: .default)

let task = session.dataTask(with: url) {(data, response, error) in //2

if error != nil { print(error!)

return }

guard let JSONdata = data else { return }

let dataString = String(data: JSONdata, encoding : .utf8) print(dataString!)

let decoder = JSONDecoder()

do{

let decodedData = try decoder.decode(MovieData.self, from: JSONdata) self.movieData = decodedData

DispatchQueue.main.async{ self.table.reloadData()

}

}catch { print(error)

}

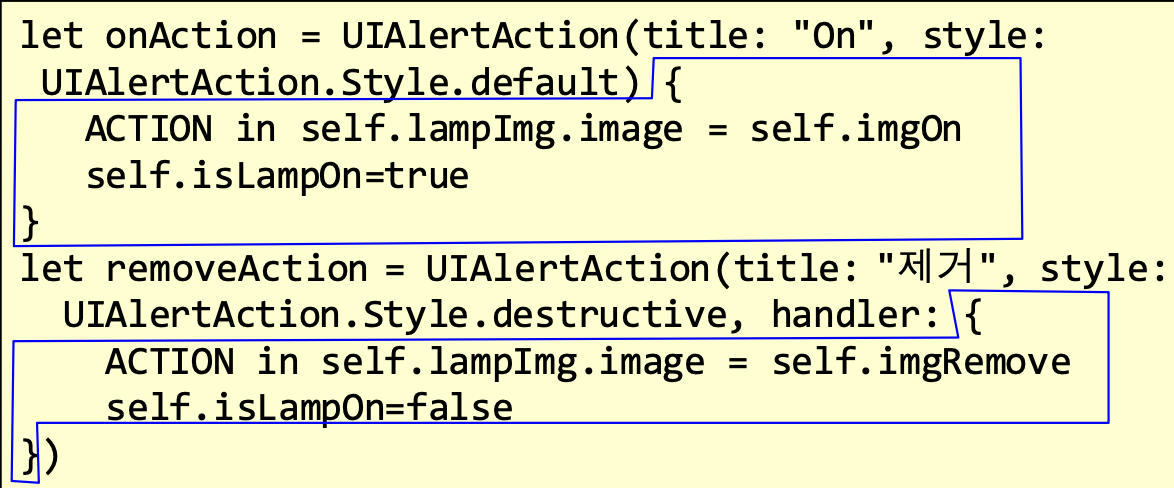
} //2

task.resume() } //1

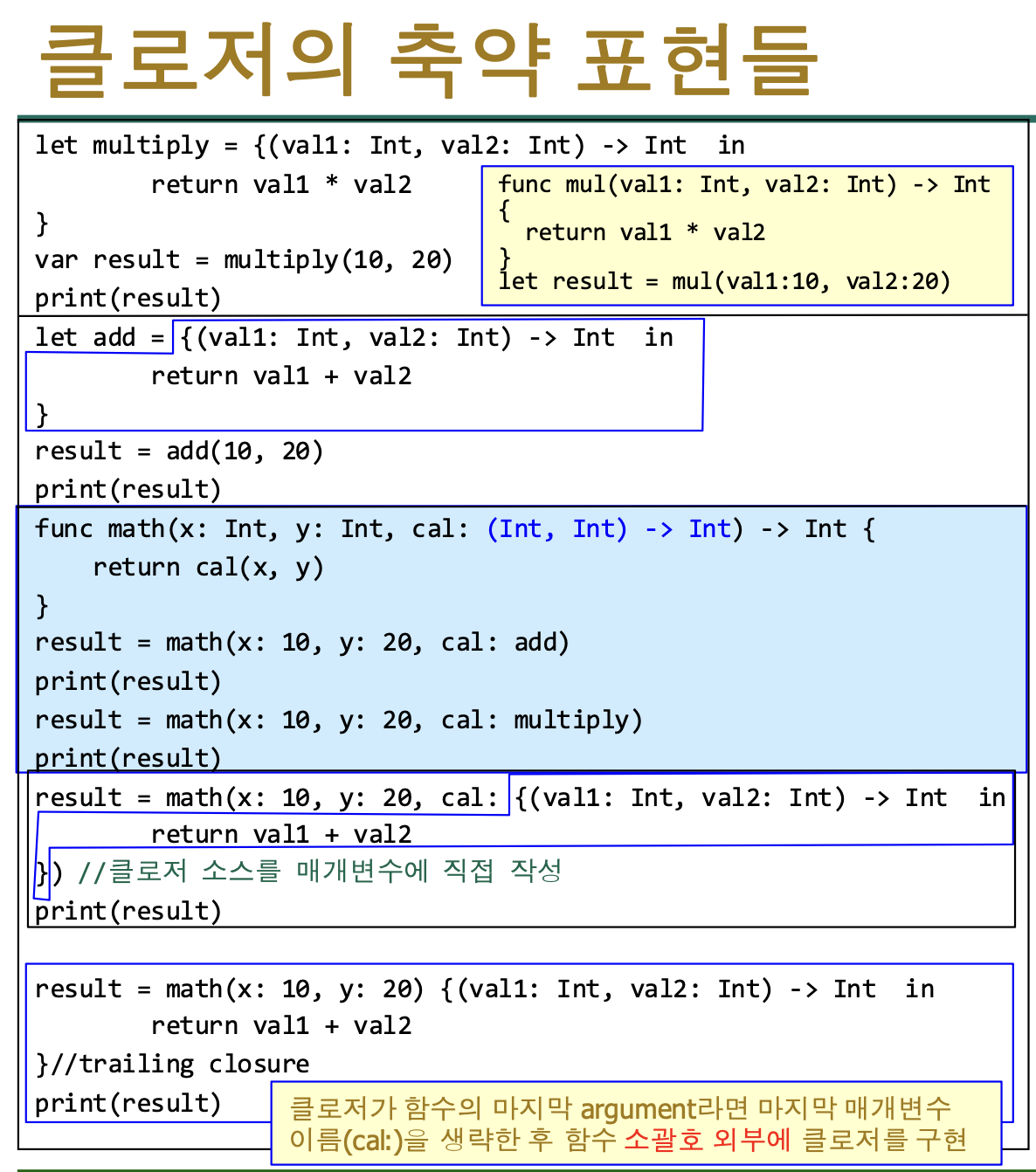
**실습.3**

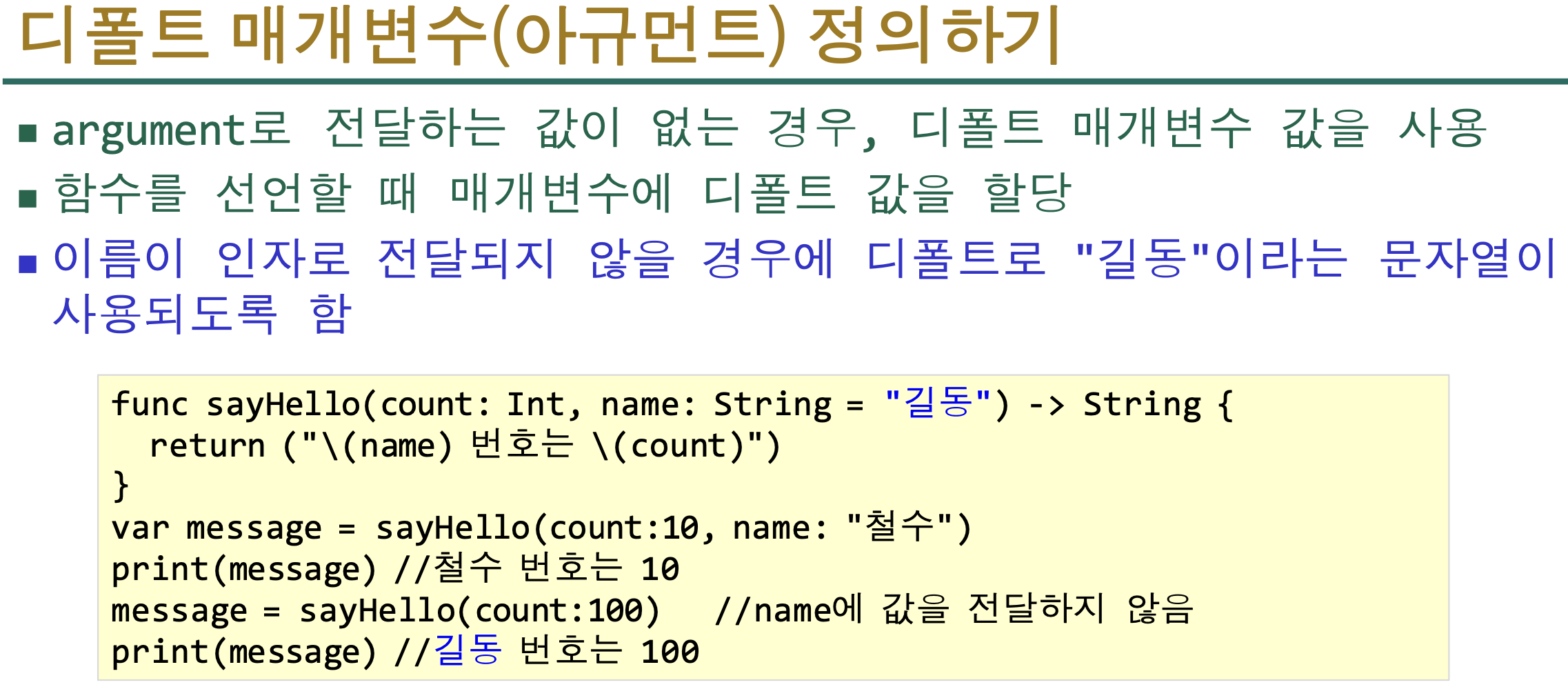
**클로저가 함수의 마지막 argument라면**

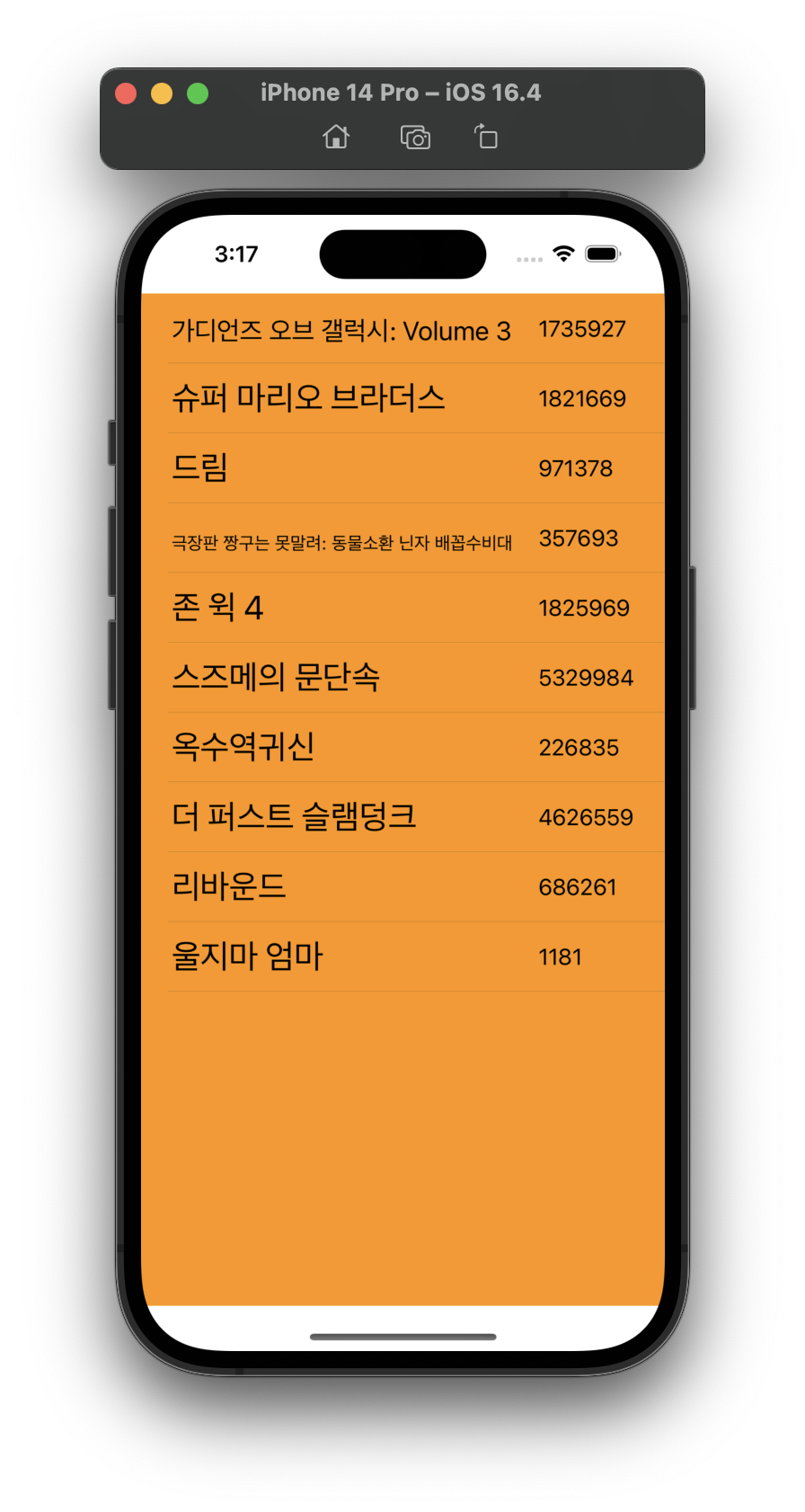
**마지막 매개변수 이름(handler:)을 생략한 후**

**함수 소괄호 외부에 클로저를 구현   
**

**실습.4**

**소스를 이해하는데에는 이 정도면 충분**

**실습.5**

****