

# Functional Programming

2015.6.13

양수한

# 차례

- 함수형 프로그래밍이란?
- 함수형 프로그래밍 장점
- 명령형 프로그램 vs 함수형 프로그램
- OOP 개발자의 전환
- 왜 함수형 프로그램인가?
- 참고

# 함수형 프로그래밍이란?

- 함수형 프로그래밍 패러다임은 순수하게 함수적인 접근 방법으로 문제 해결하는 것을 지원하기 위해 만듦
- 함수형 프로그램은 선언적 프로그래밍의 형태로 표현
- 특징 2가지
  - First-Class Function 일급 함수
  - Pure Function 순수 함수

# 일급 함수 First-Class Function

- 언어에서 제약없이 다룰 수 있는 대상
  - 함수를 정의하고 호출
  - 함수 리터럴
    - $(x: \text{Int}) \Rightarrow x + 1$
  - 이름없이 리터럴로 표기해 값처럼 주고 받음
- 함수값
  - 함수값은 객체이기 때문에 원하면 변수에 저장
    - $\text{val increase} = (x: \text{Int}) \Rightarrow x + 1$
  - 함수값은 함수

# 순수 함수 Pure Function

- Side effect 없는 함수
- Referential transparency 참조투명성
  - 주어진 입력에 대해 프로그램의 의미에 전혀 영향을 주지 않고
  - 해당 입력에 대한 그 함수의 결과 값으로 대체할 수 있다

# 순수 함수 Pure Function-Side effect

- 부작용 Side effect
  - 결과를 돌려주는 것 이외의 어떤 일을 수행
    - 변수를 수정
    - 자료구조를 제자리에서 수정
    - 객체의 필드를 설정
    - 예외exception을 던지거나 오류를 내면서 실행 중단
    - 콘솔에 출력하거나 사용자의 입력을 읽어들이
    - 파일에 기록하거나 파일에서 읽어들이
    - 화면에 출력

# 함수형 프로그램의 장점

- 순수함수들로 작성하면 모듈성 증가
- 모듈성 덕분에 순수 함수는
  - 검사<sub>test</sub>
  - 재사용
  - 병렬화
  - 일반화
  - 분석 쉽다
- 순수함수는 버그가 생길 여지가 적다
- 가독성과 유지 관리 편의성이 향상
- 반복되는 개발이 쉬워짐

# 명령형 vs 함수형 프로그래밍

특성	명령형	함수형
프로그래머가 중점두는 부분	작업을 수행하는 방법(알고리즘)과 상태의 변경을 추적하는 방법	원하는 정보와 필요한 변환
상태변경	중요	존재하지 않음
실행순서	중요	중요도가 낮음
주요 흐름 제어	루프, 조건 및 함수 호출	재귀를 비롯한 함수 호출
주요 조작 단위	클래스나 구조체의 인스턴스	일급개체와 데이터 컬렉션인 함수



# OOP개발자의 전환

- 명령형/절차적 스타일로 프로그래밍하는 데 익숙
- 순수 함수 스타일의 개발로 전환하려면 사고 방식과 개발 방법을 전환
- 함수형 프로그래밍에서는 계산 문제를 데이터 컬렉션의 순수 함수 변환에 대한 계산 및 실행
- 함수형 프로그래밍에서는 상태와 변경 가능한 데이터를 사용하지 않고 대신 함수의 적용

# 왜 함수형 프로그래밍인가?

- Multi Core
  - 동시성
    - 많은 개발자들 어려워 함
    - 사용하는 언어가 더 복잡하게 함
      - Side effect에 기반한 객체지향 프로그램
      - 멀티쓰레딩에 효과적으로 대응 어려움
- Simple
  - 함수형 프로그래밍 언어의 표현
  - 코드 이해하기 쉬움
  - 유지보수 쉬움

# 참고

- Functional Programming in Scala
- Programming in scala
- Programming scala
- [https://wiki.haskell.org/Functional\\_programming#What\\_is\\_functional\\_programming.3F](https://wiki.haskell.org/Functional_programming#What_is_functional_programming.3F)
- <https://pragprog.com/magazines/2013-01/functional-programming-basics>
- <https://msdn.microsoft.com/ko-kr/library/bb669144.aspx>
- <https://medium.com/@jooyunghan/함수형-프로그래밍-소개-5998a3d66377>
- <http://changsuk.me/?p=1916>
- <http://cafe.elharo.com/programming/java-programming/why-functional-programming-in-java-is-dangerous/>