Functional Programming

Functional Programming

정의

- Immutable state.
- No side-effects.
    - Side-effect : 반환값과 상관없는 일들.

Pure Function

"A ⇒ B" 관계로 B타입의 값에 의해서만 A타입의 값이 결정됨. 즉, 입력으로 결과를 반환하는 일 이외에 하지 않는 함수.

그 이점으로

- 추론하기가 쉽다. 예를 들어 List ⇒ Map 이라면, 결과값은 Map이고 List의 모든 요소들은 각 Map의 요소에 Mapping 된다는 의미를 알 수 있다.
- Modularity 증대. 이는 각 함수들이 독립적이고 재사용할 수 있는 Component들로 구성되고, 합성 가능(Composable)한 의미를 갖는다.

Referential Transparency

어떤 표현식(expression)을 그 평가(evaluation) 결과로 바꾸어도 프로그램의 의미가 동일한 것. 표현식(expression)의 항등원. 이는 반환값에 불변(Invariant, Immutable state)을 강제한다. 그 대표적인 예로 String, StringBuilder. String을 불변적이므로 원본 문자열에 수정을 하면 새로운 문자열이 생성되어 참조에 투명하지만, StringBuilder는 가변적이므로 원본 문자열이 변경되므로 참조에 투명하지 못함.

From:
https://wiki.ledyx.xyz/ - **Ledyx WIKI**

Permanent link:
https://wiki.ledyx.xyz/doku.php?id=base:funtional_programming

Last update: **2021/02/07 05:38**