

11. Lisp began as a pure functional language but gradually acquired more and more imperative features. Why?

- 사용되는 목적 다양화 : Lisp은 초기에 수학적 계산, 자선여치의 등 학술적 목적으로 쓰였고, 그 이를 위해 개발된 언어였지만 회사 등에서 실무적 용도로 쓰이게 되면서 사용성을 증대시키기 위해 imperative한 기능) 추가되었다.
- 효율성 증가 : 순수 함수형 언어는 프로그램을 짤 때, 함수 호출과 재귀 등으로 구현하게 되는데, 함수를 호출할 때마다 드는 리소스 소비, 시간적 손해를 줄이기 위해서 imperative한 기능이 추가되었다고 볼 수 있다. 또한 함수 호출로만 구현할 때보다 코드의 작성도 수월해졌다.

12. Describe, in your own words, the concept of orthogonality in programming language design.

Orthogonality는 직역하면 직교성으로, ① 두 개의 기능이 겹치지 않고, 예외가 없음을 의미한다.

- ① 두 개의 기능이 겹치지 않는다는 것은 (언어에서의 for, while이 같은 기능을 수행함)으로써 기능이 중복되는 것과 반대의 의미이고,
- ② 예외가 없다는 것은 (언어의 함수에서 array type은 return이 되지 않고, int나 float은 return이 되는 것과 같은 현상과 같은 예외가 없다는 의미이다.

정리하자면, 하나가 바뀌어도 나머지에 영향을 주지 않는 것이다.

직교성 (Orthogonality)가 높을수록, 변수의 type마다 연산자를 다르게 써야 하는 것과 같은 현상이 줄어들어, 개발하는 데 생산성이 향상되는 장점이 있을 수 있다.

14. What are the arguments both for and against the idea of a typeless language?

- 장점: 다양한 자료형들에 대해 배우는 학습량이 줄어들어, 언어를 더 배우기 쉽게 만들 수 있다.

또한, 개발 도중에 type의 변경이 생기거나 새로운 기능을 도입할 때 이를 이식하고 프로그램을 확장하기에도 용이하다.

- 단점: Type이 없어서 개발 중 발생한 오류를 쉽게 catch해내기 어려워지고, 코드를 검토할 때 있어서도 각 변수의 type이 없어서 변수간의 구분이나 의미 분석이 어려워져 코드 가독성을 저하시킬 수 있다. 형식은 함께 있어서도 코드의 선명도가 낮아 유지보수가 힘들어질 수 있다.