

목차

1. 트리 구조

A. 노드

B. 링크

C. 추상화

1. 트리 구조

데이터는 구조를 가진다. 구조란 우선 하나 이상의 구성 원소가 있어야 한다. 그리고 원소 사이의 연결 규칙이 있는데 그것을 구조라 한다. 즉, 데이터를 연결하는 규칙을 구조라 한다.

그렇다면 연결을 하는 이유는 무엇일까? 바로 길을 찾아가기 위해서다.

데이터 구조에는 여러 가지가 있는데 그 중 가장 많이 사용되는 것이 트리 구조다. 트리 구조는 가장 위에 위치한 노드인 루트 노드가 한 개만 존재하는 구조를 의미한다. 또한 부모 노드가 하나인 구조다.

우리가 만드는 프로그램 또한 트리 구조로 설계되었는데, 이 프로그램에서 트리의 depth는 3이고 강좌를 찾아가기 위한 인덱스는 파일 이름이다

트리 구조의 가장 큰 특징은 리니어 구조보다 탐색 속도가 빠르다는 것인데, 리니어 구조는 N만큼의 탐색 시간이 걸린다면 트리 구조는 $\log N$ 만큼의 탐색 시간밖에 걸리지 않는다.

A. ¹노드

노드란 트리를 구성하는 기본 원소로서 데이터를 저장하기 위한 단위다. 노드에는 두 가지 종류가 있다. 바로 terminal node와 non-terminal node다.

A. terminal node

terminal node는 단말 노드로 자식이 없고 더 이상 갈 길이 없는 끝 노드를 의미한다. 따라서 링크를 가지지 않는다.

B. non-terminal node

non-terminal node는 간노드로 자식이 존재하고 다른 노드로 갈 길이 존재하는 중간 노드를 의미한다.

¹ <https://blog.naver.com/realmani/222758206588>

B. 링크

링크는 노드와 노드를 연결해주는 연결선이자 길이다. 노드를 통해 데이터를 찾아간다.

노드와 링크로 이뤄진 트리 구조를 통해 결국 컴퓨터는 데이터를 저장하고 다시 찾는 것임을 알 수 있다. 이 때 데이터를 찾을 때 단순히 찾는 것 뿐만 아니라 가공하여 값을 바꾸기도 한다. 이것을 학습이라 칭한다.

C. 추상화

무언가를 기억하기 위해서는 반복되는 비슷한 것들 혹은 동일한 것들을 뽑아내야 한다. 그렇게 뽑아낸 정보들을 우리가 재사용 할 수 있는 형태로 만들어야 한다. 즉, 추상화라는 것은 데이터의 특징을 파악하여 그것이 가진 공통 구조를 인지하고, 재사용 가능한 형태의 실제 타입으로 만들어 내는 것이다.