

목차

1. Index

A. Data Structure

i. Node

- Non-terminal node
- Terminal node

ii. Link

B. 트리 구조

C. 모듈성

1. Index

길을 찾아가기 위해서 만든 데이터를 인덱스라고 한다. 따라서 인덱스는 길을 찾기 위한 정보를 가진다.

A. Data Structure

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.

i. 트리 구조

트리 구조는 가장 위에 위치한 노드인 루트 노드가 한 개만 존재하는 구조를 의미한다.

1. 1노드

노드란 트리를 구성하는 기본 원소다. 노드에는 두 가지 종류가 있다. 바로 terminal node 와 non-terminal node 다.

A. terminal node

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

B. non-terminal node

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

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

2. 링크

링크는 노드와 노드를 연결해주는 연결선이자 길이다.

우리가 만드는 프로그램 또한 트리 구조로 설계되었는데, 이 프로그램에서의 트리의 depth 는 3 이고 강좌를 찾아가기 위한 인덱스는 파일이름이다.

B. ²모듈성

프로그램을 짤 땐 모듈성을 고려하는 것이 중요하다. 모듈성이란 컴퓨터 프로그램에서 각 구성 요소의 일부를 변경하거나 증설할 때 그 변경이 프로그램 전체에 영향을 미치지 않고 변경하고자 하는 부분만 바꿀 수 있도록 설계하는 것을 의미한다.

모듈성을 고려하여 프로그램을 설계한다면 변경이 쉽고 그로 인한 영향이 적고 유지보수가 용이하다. 프로그램 관리 또한 용이하고 설계 및 코드를 재사용 할 수 있다는 장점을 가질 수 있다.

² <https://terms.naver.com/entry.naver?docId=3532985&cid=58528&categoryId=58528>