

4장 다음에 알고리즘이다. 알고리즘 후 파이썬 실습들어간다.
코로나가 좋아지면 온/오프 병행으로 간다.

정보가 단편적으로 있다가 구조를 만들면 의미가 생긴다!
데이터에서 정보로 진행을 시키는데, 정보는 의미를 가지는 것이다.
그 의미는 어떻게 만들어지는가?
데이터는 의미적인 부분을 크게 고려하지 않는다.
사실 데이터, 정보, 지식을 칼로 자르듯이 구분할 수는 사실 없다.

도시락통의 공을 생각해보라.
무엇이 도시락통의 의미를 만들었는가?
♥라는 simbol이 그 의미를 만들었다.

구조가 의미를 표현할 수 있는 것이다.
말장난을 생각해보라.
정보의 단위인 "선", "생", "님" 을 변형하였더니 의미가 달라진다.

순서구조가 달라지면, 의미가 달라진다.
우리 인간은 의미를 "구조" 를 통하여 표현하려는 경향이 있다.
구조를 변경하는 형식이 있다.
구조를 변경하는 그 규칙을 "알고리즘" 이라고 말할 수 있는 것이다.

과학자들은 자신의 발견을 숨겨야 했었다. 그리고 그것을 기록을 해야 했었다
애너그램은 글자를 똑같은데, 그 순서를 바꿔서 못알아보게 한 것이다.
케플러 앞에서 갈릴레오가 노래까지 불러가며 놀렸다고 한다.

ㄴ + ㅏ + ㄱ = 낙
ㄱ + ㅏ + ㄴ = 간

"의미는 구조가 결정한다."

허프만 코딩이나, 시, 라임 같은 것들은 그것들의 구조를 가지고 있다는 것이다.

라임이 반복되고, 줄이 바뀌고 하는 것들을 통해 인간에게 감정과 아름다움을 찾는 것이다.

구조는 논리를 만든다.

신문, 논문 등은 구조가 다르다.

사실 논리적인 구조만 가져도 사람들이 훌륭하다고 착각한다.

SCI에 컴퓨터가 쓴 논문을 제출하여 상을 받았다.

구조를 어떻게 만들까?

다른 구조는 다른 의미, 다른 느낌을 준다.

조금 복잡하게 가면, 문제 해결을 위해서 필요한 구조들이 있기도 한다.

축구에서의 포메이션 같은 것들이다.

그 축구에서의 포메이션, 구조는 그 축구팀이 추구하는 전술 그 자체를 의미한다.

지식에는 사실적 지식, 절차적 지식, 개념적 지식 등이 있다.

절차는 개별단위인 task들의 순차적 구조이다.

우선 "단위 task" 로 나누어서 순서를 만드는 것이다.

문제 해결을 위한 구조화!

문제 현상에서 정보 요소를 추출하여 구조화시키는 것이다.

정보요소들을 끄집어 내어서 구조화를 시키는 것이다.

정보요소를 추출하는 것 만으로도 문제해결이 쉬워지는 것이다.

문제는 현재상태와 해결된 상태사이의 갭이 있는 상태

문제 해결 : 그 갭을 줄이는 일

정보요소 -> input

알고리즘 -> process

what should be -> output

문제는 input을 정의, 문제의 정의를 어떻게 할 것인가?

모호한 자연어보다 명확한 언어(프로그래밍 언어, 의사코드 등)을 사용하자.

이를 통해 알고리즘 같은 것들을 만들어보자는 것이다.

자연어는 몇가지만 명확하게 하고, 나머지는 상식공간을 이용하여 대화한다.

다른 상식공간, 코드를 가지고 있으면 힘들다.

자동차 인수합병 구조화 같은 것들을 표현하는 구조가 있더라

요소와 관계선으로 표현하면 명확하고 깔끔해 보인다.

노드(정보요소)와 연결선(관계, 흐름순서)의 관계로 일단 시작을 하는 것이다.

추상화! 문제를 볼때 복잡한 상황을 중요한 요소만 남기고 표현하겠다는 것이다.

관계도나 계층도처럼 복잡한 것에서 필요없는 정보를 최대한 제거하여 핵심만 표현하는 것이다.

소프트웨어 공학 이라는 분야가 있다.

개발 기법을 배우는 것이 아니라,

팀 구성이나, 제품은 이런 것을 사용하는 것이 좋더라 등이다.

명확한 커뮤니케이션, 표현 도구를사용하는 것이 중요하다.

Decision Table(결정 테이블)

테이블, 표로 만들어버려서 모든 요소를 고려하게 하는 것이다.

이 또한 문제 해결을 위한 구조인 것이다.