

과정으로 생각하기, 컴퓨터식으로 생각하기, 컴퓨터를 다스리는 생각법*

이광근[†]
서울대학교

2/7/2016, 8/25/2015

요 약

“Computational Thinking”을 정의하고 그 내용에 포함되는 개념들을 정리한 글이다. 소프트웨어정책연구소(spri.kr) 길현영 연구원의 요청으로 마련한 글을 바탕으로 정리한 것이다.

1 배경

컴퓨터라는 도구 덕택에 인간의 지능/본능/현실은 나날이 확장하고 있고 우리는 주변의 모든 것을 컴퓨터에게 맡겨가고 있다.

이러한 디지털 환경을 바라보는 시각을 시민들에게 형성시켜주는 것. 이것이 초중고 소프트웨어 교육의 목적이다. 마치 물리교육을 통해서 우리를 둘러싼 자연을 바라보는 시각을 형성해주는 것과 같다.

이 교육의 주요 내용에는 “과정으로 생각하기”, “컴퓨터식으로 생각하기” 혹은 “컴퓨터를 다스리는 생각법”을 포함하게 된다. 사람이 컴퓨터를 다룰 때 생각하는 방법이다. 이 교육을 통해서 시민들이 컴퓨터가 만들어가는 디지털 세상을 이해하고 그 가능성과 한계를 파악하는 시각을 얻게 될 것이다.

*Computational Thinking. 종종 “컴퓨팅적 사고”, “계산중심 사고”, 혹은 “계산적 사고”로도 번역한다. 하지만 초중고생이나 시민들에게 편하게 다가가야 할 개념에 “사고”, “컴퓨팅적”, “계산중심” 등은 너무 엄숙하고 경직된 느낌이다. 어깨에 힘을 빼자.

[†]서울대학교 컴퓨터공학부 교수. 홈페이지: kwangkeunyi.snu.ac.kr

2 뜻

“과정으로 생각하기”, “컴퓨터식으로 생각하기”란 무엇을 말하는걸까.

- 컴퓨터라는 도구를 다루는 방법은 언어로 짜여진 소프트웨어다.
 - 사람은 소프트웨어를 만들고, 컴퓨터는 소프트웨어를 실행한다.
 - 컴퓨터에게 시키려는 일을 소프트웨어로 표현해서 컴퓨터에 전달하면 컴퓨터는 그 소프트웨어에 쓰인 것을 그대로 실행에 옮긴다.
- 소프트웨어를 만들 때는 지혜가 필요하다.
 - 소프트웨어에 정확히 표현해야하기 때문이다. 컴퓨터에게 일을 시키려면 그 일을 컴퓨터가 어떻게 해야 하는 지 소프트웨어로 빈틈없이 표현해야한다.
 - 그리고 짜여진 소프트웨어는 사람도 이해하기 쉬워야 한다.
 - (1) 컴퓨터가 실행에 옮길 소프트웨어이지만 과연 우리가 원하는 일을 하도록 표현한 것인지 확인하기 쉬워야하기 때문이고; (2) 소프트웨어를 통해서 사람들은 서로의 지식을 표현하고 교류하기 때문이다.
- 이 지혜가 “과정으로 생각하기”, “컴퓨터식으로 생각하기”, “컴퓨터를 다스리는 생각법”이다.

3 내용

“과정으로 생각하기”, “컴퓨터식으로 생각하기”에는 다음의 지혜가 포함된다.

- 데이터 정리하기(data structure)

자료를 정리하는 지혜다. 정리한 모양새(구조)에 따라 일을 편하게 할 수 있게된다. 하려는 일에 따라 가장 적절한 자료 정리의 모양새가 있다. 책방에 책이 아무렇게나 꽂혀있다면 책을 찾기 힘들다. 식당 줄이 일렬이 아니라면 혼돈이 온다.
- 속내용 감추기(abstraction)

복잡한것을 다루는 지혜다. 외부에는 속구현을 알려주지말라. 속구현이 어떻게 되었는 지 몰라도 만든것을 사용할 수 있도록 한다. 자동차운전은 자동차 엔진이 어떻게 만들어졌는지, 브레이크가 어떻게 만들어졌는지 몰라도 사용할 수 있다.

- 조립식으로 생각하기(composition)

독립적인 부품들을 조립해서 전체를 만드는 지혜다. 전체가 부품들로 분해될 수 있고 부품들을 제 역할에 맞게 조립하면 원하는 전체가 된다. 전체 방법은 부분 방법들의 조립이다.

 - 계층쌓기로 생각하기(hierarchy)

조립하는 과정을 여러 계층으로 쌓아가는 지혜다. 만든것을 부품으로 활용해서 더 큰 것을 만들고, 이렇게 만든 것이 또 부품이되서 더 큰 것을 만든다. 1층에서 만들어서 2층으로 올리고, 2층에서는 다시 반복하면서 차곡차곡 올라간다. 맨 윗층의 것이 우리가 원하는 것이다.
 - 끼리끼리 포장하기(module)

비슷한 것끼리 모아서 포장해 놓고 정리해 놓는 지혜다. 많이 모여 있는 것들을 잘 정리하려면, 유사한 것을 한데 모으고 박스에 넣고 박스에 표시를 해둔다. 이삿짐 포장할 때 “거실”, “부엌” 등으로 박스를 마련해 놓고 포장하고 이사하고 짐을 풀듯이.
- 반복으로 생각하기(repetition)

같은 작업을 반복하면서 일을 마치는 지혜다. 밥을 먹는 일은 수저와 젓가락으로 밥과 반찬을 입에 넣은 후 씹어삼키는 작업의 반복이다. 같은 일을 다른 데이터를 가지고 반복하면서 일을 끝마칠 수 있다. 아는 것만 알고 갔던 곳만 가면서 일을 끝마친다.
- 귀납으로 생각하기(induction & recursion)

데이터가 귀납적으로 만들어지는 종류들이 있다. 리스트는 리스트로 만든다. 트리는 트리로 만든다. A종류의 데이터를 생각하자. A 데이터를 만드는 기초적인 부품이 있고, 다음부터는 그렇게 만든 A 데이터로 새로운 A 데이터를 만든다. 나무 가지를 보자. 왼쪽 오른쪽 나무가지들로 나무가지가 만들어져 있다. 양파를 보자. 양파를 한 껍질 벗기면 다시 양파다. 양파로 양파를 만든 것이다.
- 순서로 생각하기(sequentiality & concurrency)

일의 순서를 생각하는 지혜다. 일을 쪼개고 쪼개진 일의 순서를 생각한다. 대부분의 일에는 순서가 있다. 어떤 일을 마쳤을 때에만 그 다음 일을 할 수 있다. 이전 일의 결과가 다음 일에 꼭 필요하기 때문이다.

일의 순서가 구지 없다면 한꺼번에 진행하는 지혜다. 쪼개진 일들의 순서가 필요 없는 경우가 있다. 이전 일의 결과가 다음 일에 필요한 것이 아니라면 순서없이 동시에 진행해도 된다.

- 변하는 물건을 생각하기(state)

물건의 상태를 변화시키면서 일하는 지혜다. 컴퓨터가 일하면서 컴퓨터의 메모리 상태가 변한다. 소프트웨어를 작성할 때 그런 변화과정을 생각하면서 작성하는 지혜다. 컴퓨터 메모리에 장바구니를 구현했다고 하면, 그 장바구니라는 물건이 장을 본 물건들로 채워진다. 그러면서 장보는 일을 마친다.

- 변하지않는 값을 생각하기(value)

물건을 변화시키지 않고 일하는 지혜다. 일한다는 것은 값을 만들고 만든 값으로 새로운 값을 계산하는 과정일 뿐이다. 있는 물건을 변화시키는 명령이 아니다. 옆 부서에서 만든 메뉴얼이 있다. 우리가 가져다가 우리 부서에 맞는 메뉴얼을 만들자. 옆 부서의 메뉴얼을 복사해서 우리 부서에 맞게 첨삭한 후 새로운 메뉴얼을 만든다. 옆 부서의 메뉴얼이 바뀌지는 않는다. 3에 1을 더하는 일은 새로운 값 4를 계산하는 일이다. 3이 변해서 4가 되는 것이 아니다.

- 틀을 짜서 재사용하기(framework)

자주 쓰는 것을 재사용하기 쉽게 준비해놓는 지혜다. 지금 만든 것을 지금의 경우에만 쓰게하지 말고, 지금의 상황과는 동떨어뜨려서 일반적인 경우로 확장해서 준비해놓는다. 재사용할 때마다 사용되는 상황이 다를 것이다. 다른 점을 반영해서 경우마다 조금씩 다른 일을 할 수 있도록 준비해 놓는다. 떡살 같은 것이다. 매번 같은 모양의 떡을 만들지만 넣는 재료에 따라 다른 떡을 찍어줄 것이다.

- 실행비용 생각하기(computation cost)

컴퓨터가 소프트웨어를 실행하면서 시간과 메모리를 얼마나 쓸지 어렵잡는 지혜다. 만든 소프트웨어가 실행되면 얼마나 시간과 메모리를 소모할까. 입력의 크기에 따라 비용이 어떻게 될 지를 생각하면서 더 나은 방법을 찾는 길잡이다.

- 올바른지 확인하기(correctness)

우리가 생각한 바를 틀림없이 소프트웨어로 표현했는지 확인하는 지혜다. 컴퓨터는 소프트웨어 그대로를 맹목적으로 실행할 뿐이다. 우리가

싼 소프트웨어에 실수가 있다면 컴퓨터는 그 실수를 고스란히 드러낸다. 로켓(소프트웨어)을 잘못만들면 자연(컴퓨터)이 실행하는 그 로켓은 상승중에 폭발하는 것과 같은 이치다. 우리의 의도를 실수없이 소프트웨어로 표현하는 지혜가 있다.

□

[감사] 이 글의 초고에 많은 코멘트를 해주신 서울대학교 컴퓨터공학부 허충길 교수께 감사드립니다.