"Go To Statement Considered Harmful"에 대한 요약 및 감상보고서

201811586 김동혁

이 논문에서 goto문에 한계 약점을 명시하면서 고급 언어에서 폐지되어야 된다고 주장하고 있다.

첫번째로 goto문을 사용하지 말아야 한다고 주장하는 근거로는 분기를 지시하는 저급 제어 구조이므로 블록의 개념을 가지지 못한다는 점이다.

두번째로는 사람은 동적인 구조보다 정적인 구조를 이해하는데 더 뛰어나다는 점을 뽑았다. 반복문을 사용한다면 재귀적인 형태로 보다 쉽게 구조를 이해하는 것이 가능하지만 goto문은 순서와 다르기 때문에 흐름을 복잡하게 만들어 실행의 흐름을 파악하는 것이 프로그램서를 하여금 혼란을 줄 수 있다.

세번째로는 재귀적인 절차(for, while)으로 인해 보다 편하게 표형 할 수 있고, 반복문에서의 진행 상황을 설명하는데 독립적인 좌표(for애서의 i)를 사용할 수 있다는 점을 뽑았다. 독립적인 좌표에 있어서 목표 인덱스도 중요하지만 그 외에 인덱스도 진행사항을 파악하는데 중요한 역할을 한다(goto문도 counter를 두어서 비슷한 과를 가질 수 있지만, 완전히 도움이 되지 않는다고 주장하고 있다.)

또한 goto문은 인덱스의 값이 프로그래머의 제어 범위를 벗어나지 못한다는 점도 단점으로 뽑았다

위 논문을 이해하면서 다익스트라가 프로그래머의 가독성 높은 코드를 위해서 구조적 프로그래밍을 강조한 느낌을 잘 받았다 하지만 여기서 goto문이 시간 축(위에서 아래로 진행하는 코드)을 무시했을 때 문제(스파게티 코드)가 생기는 건데 하향식 구조로만 사용한다면, 상당히 강력한 기능으로서 작용할 수 있을 거 같았다(중첩된 구조를 가진 반복문, 조건문을 한 번에 탈출할 때 강력함)

작성한 위 요소들을 종합했을 때 구조적 프로그래밍의 특징을 몇가지로 정의할 수 있는데 우선 프로그램을 읽고 이해하기 쉬워야 한다는 것이고, 프로그램의 개발 및 유지, 보수의 효율성을 높일 수 있어야 한다, 프로그래밍에 대한 신뢰성을 가질 수 있어야 한다, 프로그래밍에 대한 규칙이 제공되어야 한다, 프로그래밍에 소요되는 시간과 노력이 감소되어야 한다, 하향식으로 프로그래밍 되어야 한다, 단일 입출구를 가져야 한다는 특성을 가질 수 있을 것이다.