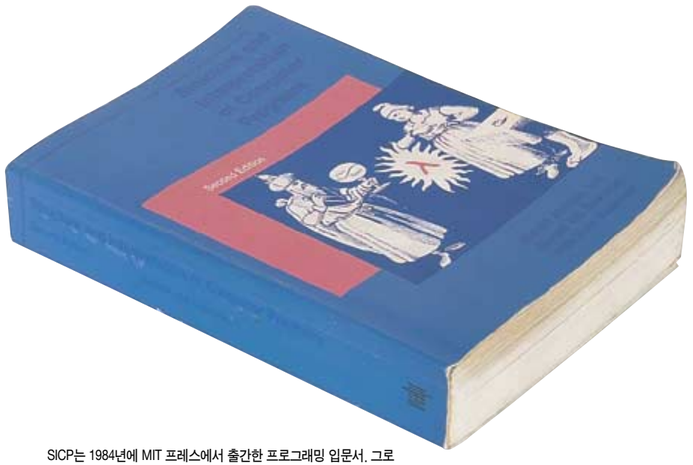
프로그래밍의 미와 스타일 구성 감각을 갖춘

# 좋은 프로그래머를 만드는 마법서

필자가 SICP(Structure and Interpretation of Computer Programs)를 처음 본 것은 대학생 시절이다. 당시 이 책이 신입생에게 프로그래밍을 가르치는 교재로 사용되고 있었기 때문에 강의를 들으며 공부할 기회가 있었다. 지금이야 그 기회가 큰 복이었다고 생각하지만 SICP의 가치를 깨닫기까지는 그저 황당한 경험일 뿐이었다. 처음 접했을 때 이 책의 내용은 독특하다 못해 고집스러워 이해하기가 쉽지 않았기 때문이다.



SICP는 1984년에 MIT 프레스에서 출간한 프로그래밍 입문서. 그로부터 12년이 지난 1996년에 두번째 판이 나왔으니, 프로그래밍을 주제로 삼아 이렇게 오랫동안 읽혀온 책도 드물다. 그야말로 '프로그래밍 과학' 분야에 고전이라 할만한 책인데, 알아보지는 않았지만 국내에서는 그리 큰 인기를 얻지 못한 것 같다.

그 당시 한 가지 더더욱 이해가 가지 않는 것은 SICP가 채택한 이상한 프로그래밍 언어였다. 스킴(Scheme)이라 불리는 이 언어는 LISP의 방언이라는데, 그동안 써왔던 다른 언어와는 너무 달라서 이 언어로 뭐 쓸만한 프로그램을 만들 수 있을까 의문이 들 정도였다. 스킴이란 언어를 배우는 데는 정말 하루가 걸리지 않았다. 그 때까지만 해도 작고 가벼운 언어는 아무래도 초보가 쓰는 게 아니겠냐는 생각을 옳다고 믿었기 때문에 스킴은 충분히 무시해도 될만한 장난감 같았다. 이듬해 다시 SICP 수업을 듣게 됐을 때 비로소 이 책의 목적이 지금까지 봐왔던 것과는 바탕부터 다르다는 점을 알게 됐다. 좁은 생각을 가두고 있던 어설픈 벽은 SICP에 의해 여지없이 무너진 것이다.

## '근본이 다른' 프로그래밍 입문서

SICP는 프로그래밍 입문서다. 난이도의 적정성에 대해서는 예전부터 논란이 많았지만 처음으로 프로그래밍을 배우는 이들을 깊이 배려해서 쓴 책이라는 점에는 재론의 여지가 없다. 다만 그 '프로그래밍'에 대한 생각이 흔히 다른 책과 아주 많이 다를 뿐이다. 예를 들어 '얼마 안에 어떤 언어 배우기' 식의 책과는 아예 근본이 다르다. 또 특정 언어나 이론을 초급부터 고급까지 단계별로 집중 연습시키는 책은 더더욱 아니다(사실 대부분의 대학 프로그래밍 입문은 이런 방식을 따른다).

SICP는 종합적인 책이다. 패러다임도 하나만 쓰지 않는다. 상태를 가진 객체, 함수, 논리, 관계, 병행성, 비결성, 소극계산법 등 여러 계산 모델을 총동원한다. 취급하고 있는 응용분야도 다양하다. 수치해석, 심볼 연산, 디지털 회로 시뮬레이션, 인터프리터, 그래픽스, 컴파일러 등을 총 망라한다. 그러나 내용이 많아 수박 겉핥기로 지나쳤을 거라고 생각하면 오산이다. 하나하나를 전문으로 다루는 교재에 비하면 덜하겠지만 각각의 아이디어와 이치에서 핵심은 심도 깊게 다뤄지고 있고 모자라는 부분은 교재와 유기적으로 결합된 예제를 통해 배우는 이 스스로가 도전하고 보충할 수 있도록 안배해뒀다. 무엇보다 이 모든 주제가 한 가지 목적을 달성하기 위해 묵직하게 흘러가기 때문에 산만하지 않은 점은 다른 책에서는 찾아보기 어려운 이 책만의 장점이다.

**김재우** kizoo@bluette.com

블루엣 인터내셔널의 기술 이사로 재직중이며 최근 본지 보물찾기 코너에서 진보하는 함수형 언어 Haskell을 연재한 바 있다.

마음에 드는 서평을 엄선해서 스크랩했다. 이어지는 서평을 읽노라면 마치 뉴스그룹에서 첨예한 의견 대립을 보이는 격렬한 쓰레드를 읽어 가는 듯 하다. 크게 엇갈린 의견은 그칠 줄을 모르고 1996년부터 2002년까지 이어지고 있다. 어떤 이는 '기념비적 책'이라고 찬사를 아끼지 않는가 하면, 다른 이는 '혼하고 쓸데없는 말을 주절대는 책'이라며 더할 수 없는 혹평을 해댄다. 아마 이 글을 읽고 SICP에 도전하는 독자가 있다면 상반된 양극의 견해로 대립하게 될지 모른다. 그러나 제대로 공부한다면 다음의 독자와 같은 경험을 하게 될 가능성이 크다.

*"학생시절에는 이 책을 정말 싫어했습니다. 스킴을 배워야 한다는 것도 끔찍했습니다. 그러나 지금은 이 책을 아주 좋아합니다. 프로그래밍에 관해 아주 깊이 있는 얘기를 담고 있기 때문입니다. (중략) 학생시절 이 책을 본 다음에, 절대 팔아버리지 마십시오. 꼭 보관하세요. 그리고 4, 5년 정도 프로그래머로 활동한 후에 다시 한 번 읽어보세요. 이번에는 틀림없이 좋아하게 될 것입니다." (A reader from Cambridge, Massachusetts, 1999 8월 19일)*

*"이 교재가 쉽지는 않습니다만 MIT나 버클리에서 입문서로 쓴다고 해도 놀랄 필요는 없습니다. 물론 정말 열심히 노력해야 이 교재의 수준과 진도를 겨우 따라 잡을 수 있겠지만 고생에는 그만한 보상이 따라 옵니다. 결국 이 책은 독자로 하여금 이론과 실제, 양면에서 광범위한 주제를 철저하게 다뤄보도록 만듭니다. (A reader from Berkeley, CA, 1999년 5월 11일)"*

이 책은 초보자만 그런 보상을 얻는 것은 아니다. 분명히 입문서인데 세월이 지날수록 경험 많은 엔지니어에게 더 많은 감명을 주는 책이다. 점차 저자의 의도가 더 잘 이해되고 경험으로 공감하게 되는 진리가 많아지기 때문에 다시 볼 때마다 미처 찾지 못했던 내용을 새로 발견하는 기쁨이 있다.

## SICP를 읽기 전에

이 책에 대해 부정적인 견해를 갖고 있는 사람도 적지 않다. 대부분 소문을 듣거나 추천받아 이 책을 보게 된 경우로 스스로 찾고자 하는 것이 무엇인지 몰라 애꿎은 책을 나무라는 경우라 할 수 있다. 그러나 학습 목적에 맞는 질 좋은 책을 골라보는 것은 독자의 책임이다. 이제 서평 속에 숨어 있는 충고를 들으면서, 이 책의 성격이 자신의 학습목적에 맞는 지 미루어 짐작해 보자.

*"코딩과 프로그래밍의 차이점. 그냥 코드를 쓰고 싶을 뿐이라면 이 책을 읽지 말라. 그런 독자는 이 책을 이해하지 못할 것이고, 오로지 불평만 늘어놓을 것이다. 대신에 '21일안에 C++ 자습하기' 식의 제목이 붙은 많은 책 중에 하나를 골라 읽는 게 낫다. (Paul G. Brown from Berkeley, CA)"*

SICP는 오로지 '프로그래밍이 무엇인가'를 제대로 소개하고 연습시키는 데 충실한 책이다. 프로그래머로 하여금 쓸만한 프로그램을 만들도록 구체적인 문제의 공략법을 알려주는 것은 이 책의 목적이 아니다. 좋은 프로그래머가 되고 싶은 이에게 가장 중요한 소양과 주어진 문제의 해결책을 프로그래밍 언어로 잘 표현해내는 능력을 스스로 키울 수 있게끔 프로그래밍의 사고를 제대로 갖춘 프로그래머를 길러 내는 책이다.

사실 SICP가 어떤 프로그래밍 책인지 설명하기는 정말 어렵다. 그만큼 논란의 여지가 많은 책이다. 그래서 직접 읽어보는 것으로도 이 책을 논할 자격은 불충분하다. 필자 역시 이 책의 가치를 이러쿵저러쿵 산만한 말로 어설프게 묘사하고 싶지는 않아 여러 사람의 서평을 빌어 SICP의 가치를 설명하고자 했다. 이제 '호그와트'로 가는 기차를 탈지 말지는 독자의 결정에 달려 있다.