

# 03

장

## 이름, 유효 범위, 바인딩

### 3.10

### 탐구문제

- 3.34** 심볼릭 디버거가 접근할 수 있는 객체와 실행 파일에 심볼 테이블 정보를 포함시키기 위해 유닉스 컴파일러와 어셈블러가 사용하는 `.stabs` 지시어를 공부해보자. 시스템 프로그래머가 어셈블리어 파일에 있는 지시어를 이해하는 데 도움이 될 수 있는 간단한 설명서를 작성하라.
- 3.35** 프로그램이 실행 시간에 자기 자신의 심볼 테이블을 조사하고 추론하게 해주는 자바, C#, 프롤로그, 펄, PHP, Tcl, 파이썬, 루비의 반사 기법을 알아보자. 이러한 기법들이 얼마나 완전한가? (예를 들어 프로그램이 현재 유효 범위에 있지 않는 기호를 조사할 수 있는가?) 반사는 어떤 경우에 좋은가? 어떤 식으로 반사를 사용하는 것이 좋거나 혹은 나쁜 프로그래밍 관습인가?
- 3.36** 펄의 `typeglob` 기법에 대해 알아보자. 이 기법은 프로그램이 자신의 심볼 테이블을 실행 시간에 수정할 수 있게 해준다. `typeglob`는 어떤 경우에 좋은가? (13장의 13.4.3절에 있는 펄에서의 `typeglobs`에 대한 보충설명에서 시작해 보자)
- 3.37** 변수가 한 파일에서 내보내지고 다른 파일에서 들여오는데 두 파일에서 이 변수의 유형을 서로 다르게 선언하는 C 프로그램을 작성하라. 이 프로그램은 성공적으로 컴파일되고 링크되지만 올바르게 동작할 수 있어야 한다. 예이다, 모듈 3, C++로 같은 작업을 해보자. 어떤 일이 발생하는가?

**3.38** C++, 자바, C# 표준 라이브러리의 모듈 계층 사용에 대해 조사해보자. 각기 어떻게 구성되어 있는가? 헤더나 패키지로 나누는 정도는 얼마나 세밀한가? 여러분이 발견한 주요 차이점에 대해 설명할 수 있는가?