

# 03

장

## 이름, 유효 범위, 바인딩

### 3.9 연습문제

3.20 르블랑-쿡 방식 심볼 테이블을 가정하고 컴파일러가 `my_firm^.revenues` [1999]와 같이 복잡한 참조의 심볼 테이블 정보(예를 들어 유형)를 어떻게 찾는지 설명하라.

3.21 다음과 같은 참조 환경을 나타내는 르블랑-쿡 방식 심볼 테이블의 내용을 보여라.

(a) 그림 3.4의 함수 `F1`

(b) 그림 3.8의 프로시저 `pop`

3.22 다음과 같은 프로그램의 실행 동안 참조 환경 `A` 리스트 내용이 어떻게 변하는지 그 추적 정보를 보여라.

(a) 그림 3.10의 프로그램. 8행에서 `양`의 값을 읽었다고 가정하자.

(b) 연습문제 3.13의 프로그램

3.23 이전 연습문제를 중앙 참조 테이블을 이용해서 반복하라.

### 3.24 다음과 같이 간단한 C 프로그램을 보자.

```
void hello() {  
    printf("Hello, world\n");  
}  
  
int main() {  
    hello();  
}
```

- (a) 위 프로그램을 `tiny.c` 및 `hello.c`와 같이 분리 컴파일되는 두 파일로 나누어라. 헤더 파일 `hello.h`를 반드시 생성하고 이를 `tiny.c`에 올바르게 포함시켜라.
- (b) 위 프로그램을 C++ 코드로 다시 생각해보자. `hello` 함수를 별도의 네임스페이스에 위치시키고 `tiny.c`에 적절한 `using` 선언을 포함시켜라.
- (c) 위 프로그램을 자바로 다시 작성하되 `main`과 `hello`를 별도의 패키지에 두어라.

### 3.25 다음과 같이 약간 더 복잡한 C 프로그램을 보자.

```
int a;  
extern int b;  
static int c;  
  
void foo() {  
    int a;  
    static int b;  
    extern int c;  
    extern int d;  
}  
  
static int b;  
extern int c;
```

각 변수 선언에 대해 변수가 외부 링크나 내부(파일-수준) 링크를 가지는지, 아니면 링크를 가지지 않는지(즉, 지역) 말하라.