프로그래밍 언어 2차 과제

20181256 수학, 컴퓨터공학과 김도현

1. Some programming languages are typeless. What are the obvious advantages and disadvantages of having no types in a language?

 Typeless language는 어떤 타입이든 간에 데이터들을 어떤 변수에도 할당 할 수 있기 때문에 flexibility 가 극대화된다. 즉 프로그래머들이type들을 따로 정의하지 않아도 코드를 빠르게 작성할 수 있다. 하지만 데이터나 ,변수들을 compiler 나 interpreter 가 통제 할 수 없게 된다. 즉 데이터나 변수가 이상해도 에러를 찾기가 힘들어 프로그래머가 직접 찾아야 하기 때문에 신뢰도가 떨어지고 어렵고 다른 사람이 봤을 때 프로그래머의 의도도 이해하기 힘들 수 있다.

2. Consider the following JavaScript program: List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

 sub1() 에서는 sub1() 내부에서의 a, y ,z 변수, global 변수인 x 가 가시적인 변수이다.

sub2() 에서는 sub2() 내부에서의 a, b, z 변수, sub1() 에서의 변수인 y, global 변수인 x가 가시적인 변수이다. (sub1이 sub2를 포함하고 있다.)

sub3() 에서는 sub3() 내부의 a, x, w변수와 global 변수인 y, z 가 가시적인 변수이다.

3. Consider the following Python program. List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

 sub1() 에서는 sub1() 내부에서의 a, y, z 변수와 global 변수인 x 가 가시적인 변수이다.

sub2() 에서는 sub2() 내부에서의 a, w 변수와 global 변수인 x, y, z가 가시적인 변수이다. (global x를 이용하여 global x 의 값을 바꾸도록 가능하게 해줬다)

sub3() 에서는 sub3() 내부의 b, z변수와 sub2() 내부의 a, w 변수, global 변수인 x, y 가 가시적인 변수이다. (nonlocal a 를 이용하여 sub3()에서 sub2() 의 a를 가시적이게 했다.)

4. Consider the following skeletal C program. Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it was defined.

a. main calls fun1; fun1 calls fun2; fun2 calls fun3.

b. main calls fun1; fun1 calls fun3.

c. main calls fun2; fun2 calls fun3; fun3 calls fun1.

d. main calls fun3; fun3 calls fun1.

e. main calls fun1; fun1 calls fun3; fun3 calls fun2.

f. main calls fun3; fun3 calls fun2; fun2 calls fun1.

 a.

fun3() 의 d,e,f 변수와, fun3()을 호출한 fun2() 의 c변수와, fun2()를 호출한 fun1() 의 b 변수와, fun1()을 호출한 main()의 a 변수이다.

b.

fun3() 의 d,e,f 변수와, fun3()을 호출한 fun1() 의 b,c변수와, fun1()를 호출한 main() 의 a 변수이다.

c.

fun1() 의 b,c,d 변수와, fun1()을 호출한 fun3() 의 e,f변수와, fun3()를 호출한 fun2() 에는 따로 없고, fun2()을 호출한 main()의 a 변수이다.

d.

fun1() 의 b,c,d 변수와, fun1()을 호출한 fun3() 의 e,f변수와, fun3()를 호출한 main()의 a 변수이다.

e.

fun2() 의 c,d,e 변수와, fun2()을 호출한 fun3() 의 f변수와, fun3()를 호출한 fun1() 의 b 변수와, fun1()을 호출한 main()의 a 변수이다.

f.

fun1() 의 b,c,d 변수와, fun1()을 호출한 fun2() 의 e변수와, fun2()를 호출한 fun3() 의 f 변수와, fun3()을 호출한 main()의 a 변수이다.

5. Consider the following program, written in JavaScript-like syntax. Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last subprogram activated? Include with each visible variable the name of the unit where it is declared.

a. main calls sub1; sub1 calls sub2; sub2 calls sub3.

b. main calls sub1; sub1 calls sub3.

c. main calls sub2; sub2 calls sub3; sub3 calls sub1.

d. main calls sub3; sub3 calls sub1.

e. main calls sub1; sub1 calls sub3; sub3 calls sub2.

f. main calls sub3; sub3 calls sub2; sub2 calls sub1.

a.

sub3() 의 a,x,w 변수와, sub3()을 호출한 sub2() 의 b,z변수와, sub2()를 호출한 sub1() 의 y 변수와, sub1()을 호출한 main()에서는 따로 없다.

b.

sub3() 의 a,x,w 변수와, sub3()을 호출한 sub1() 의 y,z변수와, sub1()를 호출한 main()에서는 따로 없다.

c.

sub1() 의 a,y,z 변수와, sub1()을 호출한 sub3() 의 x,w변수와, sub3()를 호출한 sub2() 의 b 변수와, sub2()을 호출한 main()에서는 따로 없다.

d.

sub1() 의 a,y,z 변수와, sub1()을 호출한 sub3() 의 x,w변수와, sub3()를 호출한 main()에서는 따로 없다.

e.

sub2() 의 a,b,z 변수와, sub2()을 호출한 sub3() 의 x,w변수와, sub3()를 호출한 sub1() 의 y 변수와, sub1()을 호출한 main()에서는 따로 없다.

f.

sub1() 의 a,y,z 변수와, sub1()을 호출한 sub2() 의 b변수와, sub2()를 호출한 sub3() 의 x,w 변수와, sub3()을 호출한 main()에서는 따로 없다.