CS 350: Programming Language Design

Lecture 20

Nested Subprograms, born in ALGOL 60 and iterations following and children of it both allow this feature. The purpose is to create a hierarchy of both logic and scopes.

This feature can isolate subprograms away from the rest of programs, potentially hiding them from areas where they are less useful. Static scoping is usually implemented thus grants access to nonlocal variables in enclosing subprograms.

Parameter passing methods are the ways in which parameters are sent to and pulled from subprograms which have been activated.

* In-mode: can receive data from the corresponding actual parameter
* Out-mode: can transmit data to the actual parameter
* In/Out mode: both directions

Passing Modes

* Physically move a value
* Move an access path to a value

A number of models developed by language creators

* Pass-by-Value (in-mode)
  + Value of actual is copied
  + Methods exist that allow reference without write permission, but this is difficult
  + This requires extra memory and is costly for large values
  + Using the reference method costs more resources (indirect addressing)
* By-result
  + No value is transmitted to the subprogram
  + The corresponding formal parameter acts as a local variable
  + Its value is transmitted to caller’s actual parameter when control is returned to the caller, by physical move.
  + Require extra storage locations and copy operation
* By-value-result
  + A combo of the two methods, aka Pass-by-copy
  + Formal parameters have local storage, shares weakness with both methods
* By-reference
  + Pass the access path of the variable (pass-by sharing)
  + Memory efficient
  + Slow accesses to formal param, unwanted side effects and aliasing
  + Could our language be changed in the called subprogram?
  + In Pascal and C++, formal parameters that are addressed are dereferenced which prevents such changes.
* By-name

Implementing Parameter-Passing Methods

* In most languages parameter communication takes place thru the run-time stack
* Pass-by-reference are the simplest to implement, only an address is placed in the stack\

C is pass-by-val with by-ref available through pointers

C++ offers pointer references for by-reference

Java requires all non-objects to be by-value, no method may change them.

Object parameters are by-reference in Java

C# pass-by-value, by-ref can be used by preceding formal parameters and actual parameters with *ref* keyword.

Design Considerations for Parameters,

Efficiency and data transfer plexing (half-or full duplex) to and from or just to.

These considerations are in conflict with one another. Good programming suggests limiting opportunities for changing variables to prevent errors and make debugging easier.

Pass-by-ref is memory efficient, but is programmatically dangerous.

Type-Checking

The types of actual parameters are examined for integrity with the types of corresponding formal parameters.

FORTRAN 77 and C do not offer this

Pascal and Java require this

ANSI C and C++ put the choice on the user.

Scripting languages like JS and PHP do not require type checking

Python and Ruby do not have types and thus it is impossible to implement.

Design Issues for Functions

Are side effects allowed? Parameters should always be in-mode to reduce side effects

Most types of return values are allowed?

Most imperative languages restrict the return type.

What is the max number of return values, most can only return 1 at a time, but Ruby, ML, F#, and Python can use structures to return multiple values.

In some situations, it is convenient to be able to transmit computations, rather than data, as parameters to subprograms.

Calling Subprograms Indirectly

Used when several subprograms could be run, but the one we must is not known until runtime.

C++ accomplishes this through function pointers.

Overloaded Subprograms are subprograms that have the same name as another subprogram, but may have different parameters or protocol.

Generic Subprograms, programs that are polymorphic and can take different types and change behavior to suit them. Using Class templates in C++