

Still Garbage Collecting

The inevitable question asked during a [LISP] demo was always, "What language is it written in?" This question was useful because it could be asked at the end of an hour-long nap and still seem relevant to the discussion and presentation that occurred during the nap. In addition, all work could be evaluated quite simply based on the answer to this question, as the table below (taken from "A Manager's Guide to Dealing with Researchers") shows:

ANSWER	RESPONSE
--------	----------

C	C. That's good. Very fast and efficient.
Pascal	<raise eyebrows> A toy language, but it's easy to parse, which is good.
PL/1	<chuckle> Do they still use that?
FORTRAN	<show concern> FORTRAN is really only for math, because it can't handle recursion.
APL	No, I meant what <i>programming</i> language.
LISP	<smile as if talking to a naive child> LISP isn't for real software because it is interpreted, and it does garbage collection. In the real world, you see, we do not have time to wait while a computer collects garbage.
ProLog	<see LISP>

Then came Java. At first, many of us old-time LISP hackers were offended by Java - it does nothing that LISP can't do, so why bother with it? (In fact, this is true of most things.) However, Java does things the right way:

- First of all, Java is object-oriented, which is good.
LISP is functional, which, is something else.
- Second, Java comes with extensive libraries of functions.
LISP applications use too much memory.
- Third, Java comes with its own virtual machine.
LISP is interpreted.
- Fourth, Java is portable. LISP only runs on machines with a lot of memory.
- Finally, Java does its own memory management.
LISP uses garbage collection.

Apparently, what Java has is spin doctors.