CS 474: Object Oriented Programming Languages and Environments

Fall 2015

Second Smallalk project

Due time: 9:00 pm on Monday 10/26/2015

You are required to implement a so-called *Simple Programming Language Interpreter* or *SPLI* in Cincom Smalltalk, whose purpose is to execute programs written in the *Simple Programming Language (SPL)*. SPL has a small set if instructions (a few simple statements and control structures), but uses an awkward postfix notation for its syntax (i.e., operands first, operator last). SPLI's interface contains two main components:

1. An editor field for entering and editing SPL code (one instruction per line).
2. A widget for indicating that program editing is complete.
3. Two widgets for displaying the state of the interpreter while running the program. The program state includes (a) the currently executing SPL source instruction and (b) all identifiers used in the program and their values.

SPL has just two data structures, integer numbers and linked lists. Identifiers must be declared before they are used. Code instructions are numbered, starting at 1. Program execution always starts at address line 1. SPL has the following set of statements. Each statement occupies exactly one line.

identifier INTEGER	This statement declares an integer *identifier*, which can be used in subsequent statements.
identifier NEWL	This statement declares a list identifier, creates an empty linked list and binds the list to the *identifier*.
identifier1 identifier2 COMBINE	The value bound to *identifier1* is appended to the list bound to *identifier2*. The value bound to *identifier1* (i.e., an integer or a list) becomes the first element of the list bound to *identifier2*.
identifier1 identifier2 COPY	The list bound to *identifier1* is deep-copied and bound to *identifier2*.
identifier1 identifier2 HEAD	The first element of the list bound to *identifier1* is bound to *identifier2*.
identifier1 identifier2 TAIL	A list containing all elements but the first of *identifier1* is bound to *identifier2*.
identifier integer ASSIGN	This statement assigns (binds) an integer constant to *identifier*.
identifier CHS	This statement changes the sign of the integer value bound to *identifier*.
identifier1 identifier2 ADD	This statement adds the integers bound to the two arguments and stores the result in *identifier1*.
identifer1 positive_int IF	If *identifier1* is an empty list or the number zero, jump to instruction at line *positive_int*
HLT	Terminates program execution.

Table 1: Instruction set of SPL.

Your interface should include two buttons, one for executing a single instruction (a program step) and the other for executing the program until the HLT instruction is encountered. Finally, you must use inheritance in your implementation. One good place for inheritance is to define an abstract superclass for SPL instructions with concrete subclasses for each particular instruction. You may assume that SPL

code that you must execute is correct, that is, you need not check for syntax errors. Beware of infinite loops in SPL programs.

You must work alone on this project. You are welcome to discuss the spec and the high-level design of your project with your classmates and on the Piazza blog. However, you are not allowed allowed to discuss low-level design decision and share code with others. Do not post code publicly on Piazza. Violations to this rule will result in penalties applied to the project grade of offending parties. Your project code should be in a special package called CS474. Save all your code by filing out that package in the file xxx.st, where xxx denotes your last name. Submit the file by clicking on the link provided with this assignment. No late submissions will be accepted.