Assignment Description

In this programming assignment, you will implement the translation from the AST of IC programs to the LIR language specified on the web site and validate the translated programs using the microLIR simulator. We expect you to build upon the code that you wrote in the previous programming assignments. You are required to implement the following:

Translation of functions Your compiler should translate all of the functions in the program using the AST and the information gathered during the semantic analysis phase. You compiler should also read the file `libic.sig` when the `-L` option is specified, parse it, and add it to the main AST. The compiler should be able to distinguish between calls to user-defined functions and library functions (in `libic.sig`) and emit the correct call instructions. Of course, the compiler should avoid attempting to emit a translation for the library functions themselves, as their implementation is provided externally.

Class layouts Your compiler should maintain information for each (user-defined) class about its fields and virtual functions. This information is used to determine the offsets for accessing fields (in `MoveField` instructions) and to generate the dispatch tables. In your translation, please print in comments the offset assignment for the fields of each class.
IMPORTANT: When generating field and method offsets for a class, don't forget to account for the fields and methods of its superclass and for overridden methods.

String literals Your compiler should extract all literal strings that exist in the program and translate them to LIR string literals. Instructions using the string literals should be aware of this and use the symbolic names given to the string literals.

Runtime checks You compiler should emit code to check fragile instructions (see LIR documentation) and handlers for runtime errors that print an error message and gracefully terminate the execution. We expect you to implement this code in this assignment.

Command line invocation: Your compiler must be invoked with a single file name as argument:

$$\text{java } bin \ \langle program\text{-}filename \rangle \text{ [options]}$$

With this command, the compiler will parse the input file (and `libic.sig`), construct the AST, conduct semantic analysis (reporting any errors it encounters), and translate the program to the LIR language. Your compiler must support the command-line options specified in the previous assignments, as well as the following command line option:

- The "`-print-lir`" option: the compiler will print **into a file**—`file.lir`—the LIR program translated from `file.ic`. The file should be a legal LIR program that can be read and executed on the microLIR simulator giving exactly the same results as we intend the IC program to give.

You are also expected to design a suite of tests (IC programs) that demonstrate the correctness of your translation.

Package Structure: You will implement the IR lowering in a new sub-package `lir`.

What to turn in

You must turn in your code electronically using the instructions posted in the Moodle on the due date, including

a documentation write-up. **Please include only the source files in your submission, not the compiled class files or any other temporary files.**

As in any other large program, much of the value in a compiler is in how easily it can be maintained. For this reason, a high value will be placed here on both clarity and brevity – both in documentation and code. Make sure your code structure is well-explained in your write-up.

Turn in a document PA4-DOC.XXX (XXX can be one of txt/doc/pdf) with the following information:

- A brief, clear, and concise description of your code structure and testing strategy.

- A description of the class hierarchy in your src directory, brief descriptions of the major classes, any known bugs, and any other information that we might find useful when grading your assignment. Documented bugs will be graded more forgivingly than non-documented bugs.

- A high-level description of how you implement the IR lowering.

- Feedback. We are interested in hearing your overall thoughts about the assignment: how much time you spent on it, what was the most difficult or most interesting part, and how you think it could be made better. This part is optional.

Note: Failure to submit your assignment in the proper format may result in deductions from your grade.

GOOD LUCK!