

CSE3341 Lab3

Requirement Analysis:

- complete the interpreter for the language Core
- use Tokenizer we have built in lab1&lab2
- use same language(java) as before

What interpreter should do

- read input from two files (command line)
 - first file: the Core program to be interpreted
 - second file: data file for that Core program
- two options
 - print
 - product two strings of output
 - pretty print version of the Core program
 - when write statements are executed
 - doNotPrint
 - only print when write statements are executed

Approach: Recursive descent

number of classes: one corresponding to each non-terminal

- use the Singleton pattern to ensure that there is only one Tokenizer object
- create a Prog object then call parseProg on that object

Interpreter

- interact with the user
- decide if we need to print or not
 - print
 - doNotPrint
- Scanner to control input flow
 - Scanner **pgm** to control program to be interpreted
 - Scanner **data** to control input data
 - set tokenizer with input program Tokenizer1.set(pgm)

- Parse and pass to next level

Print with indent

- only non-terminal tag as the leftmost tag of a line need to have indent

Bug Log:

[fixed] Parser works fine. But the execution part starts from the "end" token.

- possible issue: Tokenizer1.instance() needs to be saved at each level
- solution: solved by store tokenkind in each level.

[fixed] id.parse() doesn't return correct value

- solution: we have to assign value of id.parse() to a id

Files:

src/lab3/Assign.java

Assign class implementing parse, print and execute functions for the Core language's < assign > (Java source)

src/lab3/Comp.java

Comp class implementing parse, print and execute functions for the Core language's < comp > (Java source)

src/lab3/CompOp.java

CompOp class implementing parse, print and execute functions for the Core language's < comp op> (Java source)

src/lab3/Cond.java

Cond class implementing parse, print and execute functions for the Core language's < cond > (Java source)

src/lab3/CoreInterpreter.java

CoreInterpreter class interacts with user's input and perform as the driver of the interpreter (Java source)

src/lab3/Decl.java

Decl class implementing parse, print and execute functions for the Core language's < decl > (Java source)

src/lab3/DeclSeq.java

DeclSeq class implementing parse, print and execute functions for the Core language's < decl seq > (Java source)

src/lab3/Exp.java

Exp class implementing parse, print and execute functions for the Core language's < exp > (Java source)

src/lab3/Id.java

Id class implementing parse, print, execute, set value, get value, get name functions for the Core language's < id > (Java source)

src/lab3/IdList.java

IdList class implementing parse, print, execute, read and write functions for the Core language's < id list > (Java source)

src/lab3/If.java

If class implementing parse, print and execute functions for the Core language's < if > (Java source)

src/lab3/In.java

In class implementing parse, print and execute functions for the Core language's < in > (Java source)

src/lab3/Loop.java

Loop class implementing parse, print and execute functions for the Core language's < loop > (Java source)

src/lab3/No.java

No class implementing parse, print and execute functions for the Core language's < no >
(Java source)

src/lab3/Op.java

Op class implementing parse, print and execute functions for the Core language's < op >
(Java source)

src/lab3/Out.java

Out class implementing parse, print and execute functions for the Core language's < out
> (Java source)

src/lab3/Prog.java

Prog class implementing parse, print and execute functions for the Core language's <
prog > (Java source)

src/lab3/Reporter.java

Reporter class implementing a error reporter. (Java source)

src/lab3/Stmt.java

Stmt class implementing parse, print and execute functions for the Core language's <
stmt > (Java source)

src/lab3/StmtSeq.java

StmtSeq class implementing parse, print and execute functions for the Core language's
< stmt seq > (Java source)

src/lab3/Tokenizer.java

The Tokenizer interface (Java source)

src/lab3/Tokenizer1.java

Tokenizer1 class implementing Tokenizer for the Core
language tokens (Java source)

src/lab3/TokenKind.java

TokenKind enum for the Core language tokens (Java source)

src/lab3/Trm.java

Trm class implementing parse, print and execute functions for the Core language's < trm > (Java source)

src/lab3/Utilities.java

Utilities class implementing a printer to print out certain spaces.

How to compile

Compile tokenizer in edu/3341 package by command:

```
`javac -d bin src/lab3/*.java`
```

After compilation, run the TokenizerTest class by command:

```
`java -cp ./bin lab3.CoreInterpreter coreProgram inputData print/doNotPrint`
```

Note: please replace coreProgram with core program on your computer, and replace inputData with input data file on your computer.