CP471, Winter 2020

ASSIGNMENT 2 (individual): Parsing

DUE: Friday, February 14. (10% of the final grade)

In this assignment you will continue working with the grammar given to you in A1. Your task is to implement a recursive descent predictive parser with simple semantic actions.

For this you need to first modify the given grammar turning it into LL(1) form. After this is done implement the parser.

Minimal implementation requirements:

- you implementation should provide syntactical analysis functionality implemented as function (method) `Parse` (`main` in your implementation calls `Parse`);
- it should read source program from standard input token by token using `getNextToken()` from A1 (DO NOT READ FROM A FILE!!!) and write the output to the standard output (DO NOT WRITE TO A FILE!!!);
- the output is source code with diagnostics (if syntax error occurs);
- if the source code was syntactically correct it must be pretty-printed (see discussion below);
- after parsing is done, you program should print symbol tables (one for main and one per declared function) in a readable format;
- test your implementation on test cases provided in the assignment entry on `MyLearningSpace` but also create your own tests.

What and how to submit?

Submission contains two parts: documentation presenting the grammar in LL(1) form (along with relevant FIRST and FOLLOW sets) and implementation.

- Submit documentation as PDF file in A2 drop-box on MyLearningSpace.
- Submit implementation in A2 drop-box on MyLearningSpace.
- Submit source code only in a single Java or C file. It should be possible to compile your submission with `javac` or `gcc`.

Here is a sample of syntactically correct source

```
def int
gcd(int a, int b)
if (a==b) then return (a)
fi;  if (a>b) then
return(gcd(a-b,b)) else return(gcd(a,b-a)) fi; fed; print gcd(21,15);
print 45; print 2*(gcd(21, 28) + 6).
```

and here is expected pretty-printed version

```
def int gcd(int a, int b)
  if(a==b) then
    return (a)
  fi;
  if(a>b) then
    return(gcd(a-b,b))
  else
    return(gcd(a,b-a))
  fi;
fed;
print gcd(21,15);
print 45;
print 2*(gcd(21, 28) + 6)
.
```