Heikki Pulli, 014822245

compilers

Final project

NOTE: '\n' didn't work on my system to create newline, but '%n' worked. This seems to be a java convention...

doesn't work

```
print "Hello,\neveryone!";
```

works

```
print "Hello,%neveryone!";
```

Zip file contains four directories: bin/, lib/, src/ and docs/. Bin directory contains the script to run the interpreter and it uses generated .jar file in lib directory. Src directory contains .java source codes and docs directory contains the report.

Rest of the project is in github. Run

```
git clone https://github.com/HegePI/mini-pl-java.git
```

to clone the project to your local machine and run

```
./gradlew build
```

or on windows

```
./gradlew.bat build
```

to build the interpreter.

Running the interpreter

Linux

```
miniPL/bin/app <FILE_NAME>
```

Windows

```
miniPL/bin/app.bat <FILE_NAME>
```

CFG

Note: language doesn't have '&' operator, forgot to implement...

```
stmtList => stmt";" (stmt";")*
stmt => assertStmt | assignStmt | forStmt | printStmt | readStmt | varStmt
| expr

assertStmt => 'assert' '('expr')'';'
assignStmt => ident ':=' expr';'
forStmt => 'for' ident 'in' expr '..' expr 'do' stmList 'end for;'
printStmt => 'print' ident ';'
readStmt => 'read' ident ';'
varStmt => 'var' ident ':' type [':=' expr]';'

expr => cond (('<' | ['!']'=') cond)* | bool ((['!']'=') expr)*
cond => opnd (('+' | '-') opnd)*
opnd => term (('*' | '/') term)*
term => ident | number | '('expr')'

ident => [a-zA-Z0-9_]+ // any alphanumeric string with or without
underscore
type => "string" | "bool" | "number"
bool => 'true' | 'false'
number => [0-9]+
```

AST

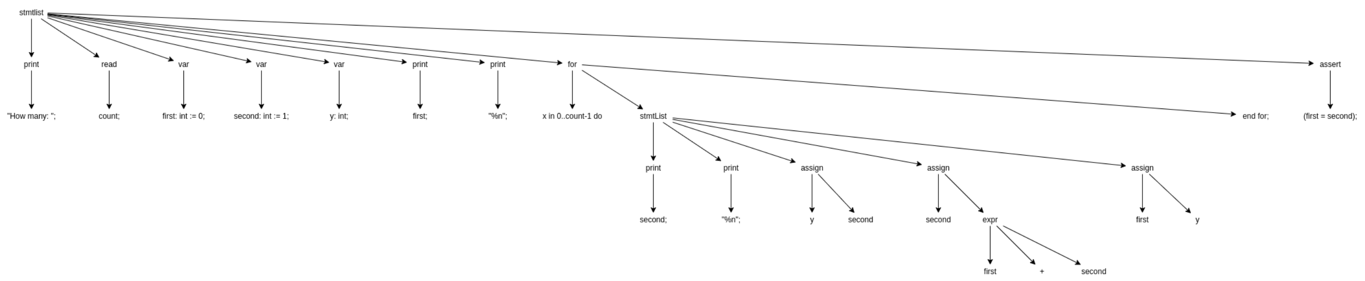
Program

```
print "How many: ";
read count;

var first: int := 0;
var second: int := 1;
var y: int;

print first;
print "%n";
for x in 0..count-1 do
    print second;
    print "%n";
    y := second;
    second := second + first;
    first := y;
end for;

assert (first = second);
```



Error handling

When encountering an error interpreter prints out either scanninError, parsingError or interpretingError depending on at what stage of execution error occurs. When error occurs interpreter prints out an error message and execution halts.

Work hours

Approximately 25-30 hours. Quite a lot of time was spent at the start, when trying to implement interpreter first in Rust programming language.