report.md 3/18/2022

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!";
```

## Building the interpreter

```
cd src/app
./gradlew build
```

generated files are located at build/ directory.

## 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
```

report.md 3/18/2022

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

#### **AST**

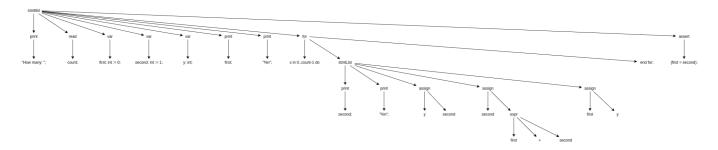
#### Program

```
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);
```



report.md 3/18/2022

# 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.