Documentation for P5

Overall, in this project, I changed ast.java, P5.java, and wrote test.crrt, typeErrors.crrt. To invoke my project, first type “make”. Then type “make test”. If there are name errors in the program, it will report them. If not, the program will do type chek. If there are type errors, it will report them. If not, the program will call unparsed, you will see the result in the corresponding file.

ast.java:

void typeCheck(Type returnType) for every statement node and fnBody.

The returnType is the return type of a function. This argument is for checking type errors in return statement. To compare the return type and the type in return statement, I need to store the return type and pass it throw the function body.

Type typeCheck() for every expression node.

For expression nodes, we need to pass it’s type up through the AST, so that we can deal with nest expressions. This also handle that we never report duplicated error by return an error type.

void typeCheck() for some high level nodes.

For high level nodes like programNode, there is no need to pass in argument nor return its type.
And no typeCheck() for declarations.

class hierarchy.

I added four subclasses for binaryExpNode: ArithmeticExpNode, RelationalExpNode, LogicalExpNode, EqualityExpNode, corresponding to +-*/, <><=>=, =!===.

New fields and methods.

For each ExpNode, I added lineNum and charNum for error message location. Also method to get these two field.

P5.java:

Added NAME_ANALYSIS_ERROR for name errors. Added type error check and lines to catch those errors.

test.crrt:

A Carrot program that thoroughly tests every possible error and can pass the type check analysis.

typeErrors.crrt:

A Carrot program that thoroughly tests every possible error. It will cause all kinds of errors.