Linden Beemer
CPSC 326
Project Plan

Goals of extension:

- Specify the syntax used in MyPL to implement error handling using the standard try catch block
- Implement this grammar across the lexer, parser, static checker, and VM
- Learn more about how try-catch blocks work behind the scenes

Planned Syntax in MyPL

My proposed syntax for error handling in MyPL will be one similar to what is seen in Java. A few example blocks are as follows:

```
var int x = nil
var int y = 10
var boolean result = nil
try {
   result = x < y
} catch StaticError e {
   result = false
}
```

```
try {
   var int myVar = new T
} catch Error e {
   print(e)
}
```

As seen, the error can be defined as a specific error, or to catch any errors thrown. There is no finally block included like Java has.

Grammar in MyPL

The proposed grammar for this block (within the confines of MyPL's current grammar) is as follows:

```
<trycatch_stmt> ::= TRY LBRACE ( <stmt> )* RBRACE CATCH <error> ID
     LBRACE ( <stmt> )* RBRACE

<error> ::= ERROR | LEXERERROR | PARSEERROR | STATICERROR | VMERROR
```

Error may be expanded as more errors are covered in this class (such as potentially runtime if possible?).

Linden Beemer
CPSC 326
Project Plan

Game Plan

In order to implement my extension, the following needs to happen in each section of the language:

- Lexer must be updated to recognize try, catch, and error types.
- Object for try/catch blocks must be added to the AST parser files
- Parsing must be updated to accurately check for try catch statements
- Pretty printer must be updated to correctly print try catch statements
- Static checker must work with updated language (luckily the parser should catch badly typed catch statements)
- VM must allow for instructions related to try catch statements
    - This is the part I have no idea how to implement
- Any extra aspects that may be touched on as we finish the MyPL assignments

Initial Test Cases

My initial test cases will be the ones above for syntax examples, as well as these basic test cases below:

```
try {} catch LexerError e {}

try {} catch ParseError e {}

try {} catch StaticError e {}

try {} catch VMError e {}

try {} catch Error e {}

try {
   var int x = 10
} catch Error e {
   print(e)
}

try {
   var x = nil
} catch StaticError e {
   print(e)
}
```

I will, of course, add more when I implement it to ensure that this extension works.