Plan for Today

Finish multiple scopes in symbol table and Type representation

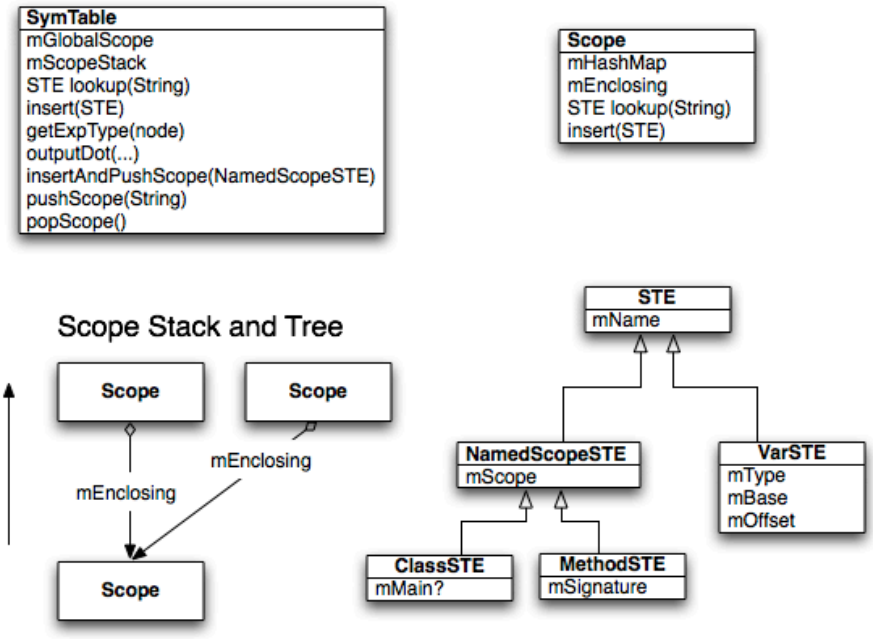
Type errors in MeggyJava

- errors relevant to methods and classes

Implementing classes

- dynamically allocating an object
- the "this" parameter and expression
- uses and defines of class member variables

SymTable, Scope, and STE classes



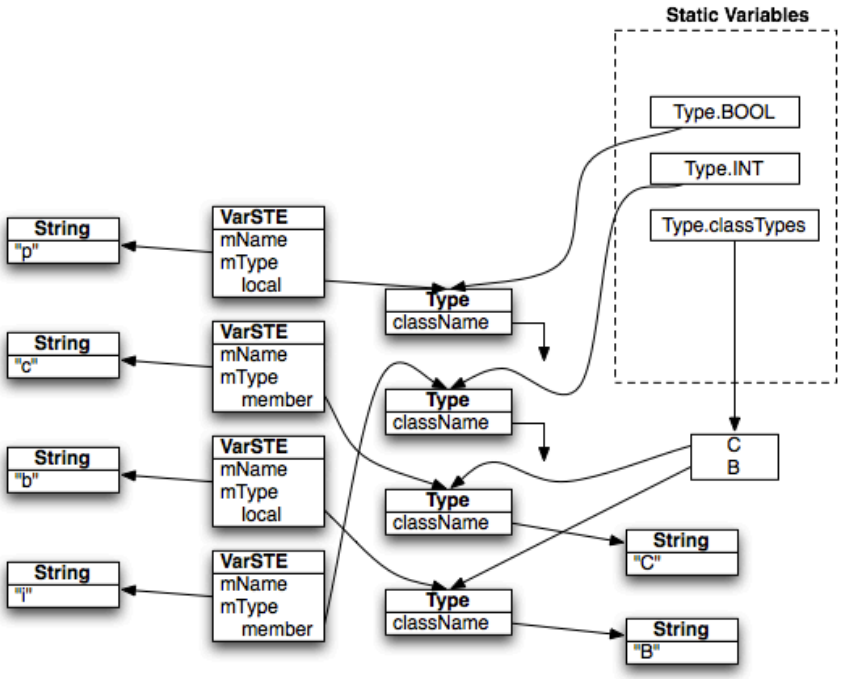
Type implementation in the MeggyJava compiler

```
public class Type {
  public static final Type BOOL = new Type();
  public static final Type INT = new Type();
  …

  // class type map (key: class name, value: type)

  private static final HashMap<String, Type> classTypes
    = new HashMap<String, Type>();
```

Only one instance of the type object per atomic type and class type

- to determine if types are equal just compare references
- Type class provided does not know about inheritance

More sophisticated type representation



Implementing type checking for PA6 MeggyJava

Visitor over AST will check for type errors at each AST node

Syntax *AST node*

`id = Exp ;` `AssignStatement(id, Exp)`

[LINENUM,POSNUM] Undeclared variable VARNAME

[LINENUM,POSNUM] Invalid expression type assigned to variable VARNAME

```
public Type name(...) {...return Exp; }
                                    MethodDecl(name, Stms, Exp)
```

[LINENUM,POSNUM] Invalid type returned from method METHODNAME

Exp . name (Args) CallExp(name, Args)

[LINENUM,POSNUM] Receiver of method call must be a class type

[LINENUM,POSNUM] Method METHODNAME does not exist

[LINENUM,POSNUM] Method METHODNAME requires exactly NUM arguments

[LINENUM,POSNUM] Invalid argument type for method METHODNAME