

JSuper.java

```
1  // Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas
2
3  package jminusminus;
4
5  import static jminusminus.CLConstants.*;
6
7  /**
8   * The AST node for a "super" expression. It serves as a target of some field
9   * selection or message.
10  */
11
12  class JSuper extends JExpression {
13
14      /**
15       * Construct an AST node for a "super" expression given its line number.
16       *
17       * @param line
18       *         line in which the expression occurs in the source file.
19       */
20
21      public JSuper(int line) {
22          super(line);
23      }
24
25      /**
26       * Analysis involves determining the super class to that in which we are in;
27       * this becomes the type.
28       *
29       * @param context
30       *         context in which names are resolved.
31       * @return the analyzed (and possibly rewritten) AST subtree.
32       */
33
34      public JExpression analyze(Context context) {
35          type = ((JClassDeclaration) context.classContext.definition())
36                  .thisType();
37          if (type.isReference() && type.superClass() != null) {
38              type = type.superClass();
39          } else {
40              JAST.compilationUnit.reportSemanticError(line(),
41                  "No super class for type " + type.toString());
42          }
43          return this;
44      }
45
46      /**
47       * Load "this" onto the stack (even if we treat it as its super class.
48       *
49       * @param output
50       *         the code emitter (basically an abstraction for producing the
51       *         .class file).
52       */
53
54      public void codegen(CLEmitter output) {
55          output.addNoArgInstruction(ALOAD_0);
56      }
57
58      /**
59       * @inheritDoc
60       */
61
62      public void writeToStdOut(PrettyPrinter p) {
63          p.println("<JSuper/>");
64      }
65
66  }
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder