

## JAST.java

```
1  // Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas
2
3  package jminusminus;
4
5  /**
6   * The abstract superclass of all nodes in the abstract syntax tree (AST).
7   */
8
9  abstract class JAST {
10
11     /** Current compilation unit (set in JCompilationUnit()). */
12     public static JCompilationUnit compilationUnit;
13
14     /** Line in which the source for the AST was found. */
15     protected int line;
16
17     /**
18      * Construct an AST node the given its line number in the source file.
19      *
20      * @param line
21      *         line in which the source for the AST was found.
22      */
23
24     protected JAST(int line) {
25         this.line = line;
26     }
27
28     /**
29      * Return the line in which the source for the AST was found.
30      *
31      * @return the line number.
32      */
33
34     public int line() {
35         return line;
36     }
37
38     /**
39      * Perform semantic analysis on this AST. In some instances a new returned
40      * AST reflects surgery.
41      *
42      * @param context
43      *         the environment (scope) in which code is analyzed.
44      * @return a (rarely modified) AST.
45      */
46
47     public abstract JAST analyze(Context context);
48
49     /**
50      * Generate a partial class for this type, reflecting only the member
51      * information required to do analysis.
52      *
53      * @param context
54      *         the parent (class) context.
55      * @param partial
56      *         the code emitter (basically an abstraction for producing the
57      *         partial class).
58      */
59
60     public void partialCodegen(Context context, CLEmitter partial) {
61         // A dummy -- redefined where necessary.
62     }
63
64     /**
65      * Perform code generation for this AST.
66      */
67 }
```

```
67     * @param output
68     *         the code emitter (basically an abstraction for producing the
69     *         .class file).
70     */
71
72     public abstract void codegen(CLEmitter output);
73
74     /**
75     * Write the information pertaining to this AST to STDOUT.
76     *
77     * @param p
78     *         for pretty printing with indentation.
79     */
80
81     public abstract void writeToStdOut(PrettyPrinter p);
82
83 }
84
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder