

JWildExpression.java

```
1  // Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas
2
3  package jminusminus;
4
5  /**
6   * The AST node for a "wild" expression. A wild expression is a placeholder
7   * expression, used when there is a syntax error.
8   */
9
10 class JWildExpression extends JExpression {
11
12     /**
13      * Construct an AST node for a "wild" expression given its line number.
14      *
15      * @param line
16      *         line in which the "wild" expression occurs occurs in the
17      *         source file.
18      */
19
20     public JWildExpression(int line) {
21         super(line);
22     }
23
24     /**
25      * Simply set the type to ANY (a wild type matching everything).
26      *
27      * @param context
28      *         context in which names are resolved (ignored here).
29      * @return the analyzed (and possibly rewritten) AST subtree.
30      */
31
32     public JExpression analyze(Context context) {
33         type = Type.ANY;
34         return this;
35     }
36
37     /**
38      * No code generation.
39      *
40      * @param output
41      *         the code emitter (basically an abstraction for producing the
42      *         .class file).
43      */
44
45     public void codegen(CLEmitter output) {
46         // Nothing to do
47     }
48
49     /**
50      * @inheritDoc
51      */
52
53     public void writeToStdOut(PrettyPrinter p) {
54         p.printf("<JWildExpression line=\"%d\"/>\n", line());
55     }
56
57 }
58
```