```java
// Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas

package jminusminus;

import static jminusminus.CLConstants.*;

/**
 * The AST node for the boolean "false" literal.
 */

class JLiteralFalse extends JExpression {

    /**
     * Construct an AST node for a "false" literal given its line number.
     *
     * @param line
     *            line in which the literal occurs in the source file.
     */

    public JLiteralFalse(int line) {
        super(line);
    }

    /**
     * Analyzing a boolean literal is trivial.
     *
     * @param context
     *            context in which names are resolved (ignored here).
     * @return the analyzed (and possibly rewritten) AST subtree.
     */

    public JExpression analyze(Context context) {
        type = Type.BOOLEAN;
        return this;
    }

    /**
     * Generating code for a boolean literal means generating code to push it
     * onto the stack.
     *
     * @param output
     *            the code emitter (basically an abstraction for producing the
     *            .class file).
     */

    public void codegen(CLEmitter output) {
        output.addNoArgInstruction(ICONST_0);
    }

    /**
     * Generating branch code for a boolean literal is trivial; it's either
     * empty or an unconditional branch.
     *
     * @param output
     *            the code emitter (basically an abstraction for producing the
     *            .class file).
     * @param targetLabel
     *            the label to which we should branch.
     * @param onTrue
     *            do we branch on true?
     */

    public void codegen(CLEmitter output, String targetLabel, boolean onTrue) {
        if (!onTrue) {
            output.addBranchInstruction(GOTO, targetLabel);
        }
```

```
67      }
68
69      /**
70       * @inheritDoc
71       */
72
73      public void writeToStdOut(PrettyPrinter p) {
74          p.printf("<JLiteralFalse line=\"%d\" type=\"%s\"/>\n", line(),
75                  ((type == null) ? "" : type.toString()));
76      }
77
78  }
79
```