Level 3: Code Generation

The next task is to generate code for the MicroJava Virtual Machine. Before you start, carefully study the specification of the VM (Appendix B) in order to become familiar with the run-time data structures, the addressing modes, and the instructions. Then do the following steps:

1. Create a new package *MJ.CodeGen*.
2. Download the files *Code.java*, *Operand.java* and *Decoder.java* from student portal into this package.
3. Complete the skeleton file *Code.java* according to the slides of the course.
4. Add semantic actions to *Parser.java*. These actions should call the methods of *Code.java* and *Operand.java* as shown on the slides. Start with the actions for selectors (e.g. *obj.f* and *arr*[*i*]), and continue with the semantic actions for expressions, assignments, if statements, while statements and method calls. Note that most context conditions from Appendix A.4 have to be checked here as well.
5. Download the file *Compiler.java* into the package *MJ*. This is the main program of your compiler that replaces *TestParser.java*. Compile it and run it on *sample.mj*. This should produce a file *sample.obj* with the compiled program.
6. Download *BuggySemanticInput.mj* and check if your compiler detects all semantic errors in this MicroJava program.

In order to run your compiled MicroJava programs download the file *Run.java* (i.e. the MicroJava Virtual Machine) into the package *MJ* and compile it. You can invoke it with

```
java MJ.Run sample.obj [-debug]
```

You can also decode a compiled MicroJava program by downloading the file *Decode.java* to the package *MJ* and compiling it. You can invoke it with

```
java MJ.Decode sample.obj
```