

## Compiler Construction 2013

### Project - part three - Optimization and Java byte code generation

- Implement arbitrary optimization techniques on your intermediate representation of Mini Java. The more, the better.
- Implement a code generation unit that transforms your intermediate representation to java byte-code. For the generation of Java byte code use the bcel library (<http://commons.apache.org/proper/commons-bcel/>).  
Hint: Use the javap tool with argument -c (javap -c foo).
- The provided test suite specifies the byte code generation rules and is mostly for you to get a feel for the number of features you have implemented. It will be used by us to do an initial evaluation of your work, but we reserve the right to use additional test cases and other methods for grading. You are not allowed to modify the test cases in any way.
- The byte code generator should reuse the front end implemented in the first and second part of the project. Feel free to improve and modify the code from the first part as long as it conforms with the test suites (first, second and the third part of the project).
- The provided test suite should be imported into your eclipse project.
- 28 days to complete this part of the project (by 17.0.2013 10:00h)
- Provide solutions (the zipped Eclipse project, including the parts provided by us) in the form cc-project-3-name1-name2.zip e-mailed to cc-staff@iam.unibe.ch. The subject of the e-mail should be cc-project-3-name1-name2. name1 and name2 refer to the names of the team members providing the solution.
- You are not allowed to use pre made files from the internet, all parts of the project solution should be the authors own work.