

Language Processors - Coursework 3

Organisation: Do this work in pairs if you wish. You will hand work in as individuals but the handin procedure will allow you to say who you collaborated with. *Pairs* only – not threesomes!

Handin:

- A zipped version of your solution directory after removing all the `.class` files (just to save space). Alternatively, you can only include the files that were modified and added by your implementation.
- A test file (MiniJava program) with all the new cases added to the language, including expressions and statements that should throw type errors. Point these expressions out (good and bad ones) by adding a comment (find out how you can add comments in MiniJava). The typechecker will report the first type error and quit. You can test your program by only including one ill-typed expression/statement at a time and commenting out the rest (that's what I will do with your file).

The work

You will extend the MiniJava implementation, including the typechecker, with two new expressions and a new statement. Copy the MiniJava implementation directory to where you want to work, using:

```
cp -R /soi/sw/courses/daveb/IN2009/minijava/chap5 .
```

Files README in the various directories give a brief description of the structure of the implementation.

1. Add the `||` (or) operator to MiniJava (implemented in CW2).
2. Add the `?:` (shortcut if-else) operator to MiniJava, where `op1?op2:op3` returns `op2` if `op1` is true, and returns `op3` if `op1` is false. The syntax is:

`<Exp> '?'<Exp> ':'<Exp>`

3. Add the repeat-until statement to MiniJava (implemented in CW2).

Your implementation should correctly typecheck the new expressions and statement. Lab6 explains how to extend MiniJava (typechecker included) with a new expression. The process is similar for statements where, instead of modifying `TypeCheckExpVisitor`, you will have to modify `TypeCheckVisitor`. In a nutshell, you will have to:

- Modify the `minijava.jj` file.
- Add the classes representing the new expressions and statement to the `syntaxtree` directory. Note that this time you will also have to include an `accept` method to be used by the Type visitor (lab6).

- Update the pretty-printer visitor by adding the new `visit` declarations to `Visitor.java` and the new implementations of `visit` to the `DBPrettyVisitor.java` file.
- Update the default Type visitor by adding the new `visit` declarations to `TypeVisitor.java` and the new implementations of `visit` to the `TypeDepthFirstVisitor.java` file.
- Implement the actual type checking of the new expressions and statement by adding the respective `visit` method to `TypeCheckExpVisitor.java` (expressions) and `TypeCheckVisitor.java` (statements).