Voortgezet Programmeren

Block 3, 2013

Exercise 0 (not graded)

Start Eclipse (Start/All Programs/Programming/Eclipse), and choose the workspace in a directory where you have write access. Create a new java project (File/New/Java project). Add a new class HelloWorld to the project (File/New/Class), and input the following main-method:

```
public static void main(String[] args) {
        System.out.println("Hello world!");
}
```

Save the class, and execute it (Run/Run As/Java application). Notice that in addition to the central text editor the default view of Eclipse also includes a bottom tab with console output, and tabs on the left side with package- and class hierarchy browsers.

Add the following lines within the main():

```
GregorianCalendar c = new GregorianCalendar();
System.out.println("The current time is" + c.getTime());
```

Now you should should see a red “x” next to one of the lines - this signifies a compilation error. Click on the “Problems” tab on the bottom pane, to find out what is wrong. Then, move the cursor to the word with red underlining, and do ctrl+1 (quick fix). Quick fix allows you to automatically import the missing package. You can also do ctrl+shift+o (organize imports) to fix all imports simultaneously.

Add code lines that declare a variable with the name squareRootOfFour and type double, assign Math.sqrt(4.0) to it, and print the value. Eclipse allows to auto-complete variable and function names with alt+/. Try auto-completing the variable name second time you type it.

Add the following lines:

```
double secret = 2.0 / 5.0 / 2.3 * 8.0 + 1.23;
secret += 2.0 / 3.0 + 4.0 - (8.0 / 124.8);
secret *= 444.3;
```

Toggle a breakpoint in the first of these lines (cursor to line, ctrl+shift+b). Run the class in debugger (Run/Debug), and change to debugging perspective when asked. Try to step over the lines (Run/Step over or F6), and notice how you can observe variable values while debugging. Afterwards change back to Java perspective (button in top-right corner).

Add desription of the class in the comments above the class: before the class signature (public class ...) type '/**' and press enter. Eclipse should now auto-complete this for a full Javadoc documentation block. Now fill in class description (first line = short description) and add also your name and student number in the Javadoc author tag (@author).