

Exercise 10: Getting started with Java programming in Linux

Hanne Munkholm <hm@itu.dk>
IT University of Copenhagen

August 18, 2005

Abstract

This exercise will not teach you how to do java programming. You should learn that in your courses. The purpose of the exercise is to introduce you to some tools for Java programming on Linux. We will look at two different java compatible editors: Nedit and Emacs, and a Java Development Environment BlueJ. You will learn to write, compile and run a simple program with either of these tools and the command line. Also try Eclipse (not documented in this exercise yet).

1 Setting up your Java environment

- The Java Development Kit installed on the student PCs at ITU is Suns jdk1.5.0_01.
- *Setting your PATH*: Check if the "javac" from Suns JDK is already in your PATH:

```
$ which javac
```

If the output is either `/usr/java/jdk1.5.0_01/bin/javac` or `/usr/java/jdk/bin/javac` your PATH is already correct. If not, open the file called ".bashrc" in your home directory with a text editor, and add the following line:

```
export PATH=/usr/java/jdk/bin/:$PATH
```

- *Setting your CLASSPATH*: Test if your CLASSPATH is already correct with the command:


```
echo $CLASSPATH
```

The output should be either `./usr/java/jdk1.5.0_01` or `./usr/java/jdk`. If this is not the case, open the file called ".bashrc" in your home directory with a text editor, and add the following line:

```
export CLASSPATH=./usr/java/jdk
```

- If you had to set your PATH and/or your CLASSPATH, log out and log in again to make sure the changes takes effect.
- NB! You might have to delete your kde preferences to make the new PATH and CLASSPATH settings take effect in programs you start from the KDE-menu.

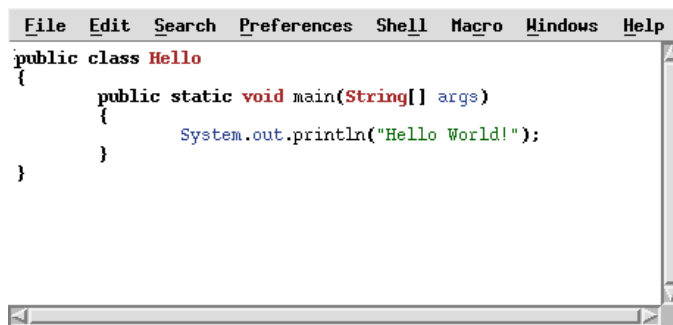
2 Nedit

- Nedit is a text editor with support for java. Nedit can be started by double clicking the  icon in Window Maker, by selecting it from the menu GNOME or KDE (“Red Hat menu → Programming → Nedit”) or by typing “nedit” at the command prompt.

2.1 Typing in your program


- Type a simple “Hello World” java program, and save it as “Hello.java”:

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



- You will notice when you have saved the file with a “.java” extension, that nedit discovers it is a java file, and enables syntax highlighting (colors your code).

3 Compile and run your program

- Open a terminal window, e.g. “konsole” in KDE, GNOME Terminal in GNOME (“Red Hat Menu → System Tools → Terminal ” or “xterm” in Window Maker (double click the  icon in the Window Maker Dock).

```
$ javac Hello.java
```

at the command prompt. If the program contains any syntax errors, they will be displayed. If there are no errors, you just get your command prompt back.


- To run your program, type

```
$ java Hello
```

in your command prompt. Your program should then display the text

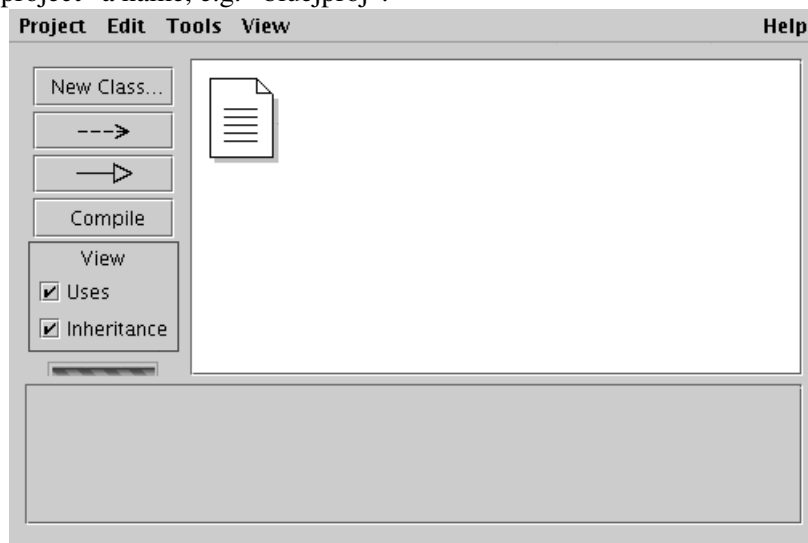
```
Hello World!
```

4 BlueJ

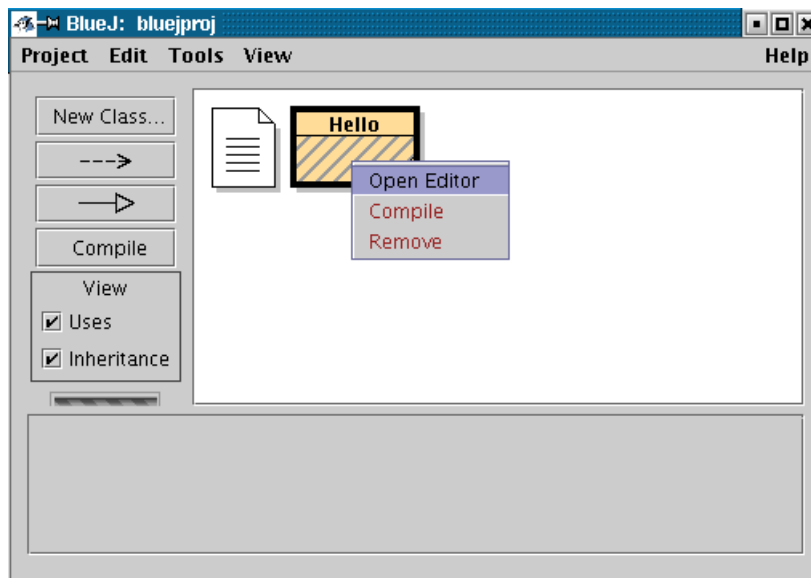
BlueJ is a graphical java development environment. Open BlueJ from the KDE-menu
→ Programming → BlueJ , or by typing "bluej" at the command prompt.

4.1 Typing in your program

- Select "Project", the "New Project" in BlueJ's menu. Give your new "BlueJ project" a name, e.g. "bluejproj".



Click Edit → New Class, type "Hello" in the "Class Name" field, and click Ok.



Right click on the "Hello" class, and select "Open Editor".

Insert the code

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

in the class, and save the file (Class → Save).

4.2 Compile and run your program

- Compile your program by clicking the "Compile" button. Or by right-clicking on the class in the GUI, or by selecting "Compile" in the "tools" menu. When the "Hello" class is compiled, it is no longer striped:



- Run your program by right-clicking the "Hello" class again. Now that it has been compiled, a few new menu items have appeared. You can start a new instance of the "Hello" class, and inspect its variables etc. This is good for testing and debugging. You can also run the "main" function:



BlueJ will now display your output ("Hello World!") in a terminal window.

5 Emacs

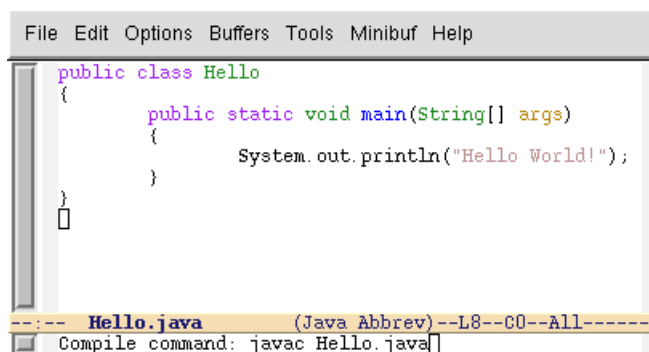
Start Emacs from the Window maker menu → Editors → Emacs, or by typing "emacs" at the command prompt. Learn basic Emacs in exercise 3.

5.1 Typing in your program

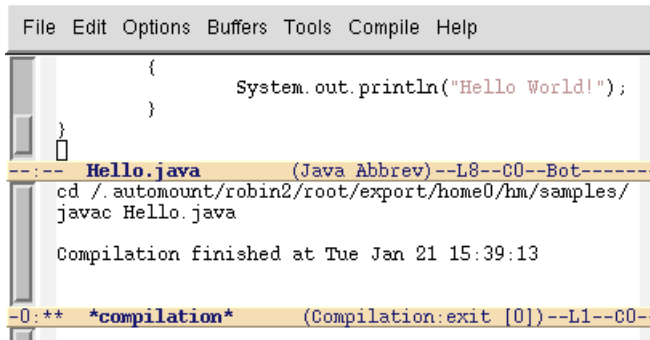
Type in the program as previously, or open the "Hello.java" file you have already made.

5.2 Compile and run your program

Click "Compile" in the "Tools" menu. Emacs will ask you for a compile command, offering "javac Hello.java" as default:

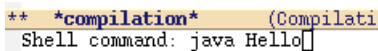


Press "enter". Emacs will split in two windows, and the output of the compile will appear in one of them.



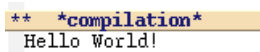
```
File Edit Options Buffers Tools Compile Help
{
    System.out.println("Hello World!");
}
Hello.java (Java Abbrev)--L8--C0--Bot-----
cd / . automount/robin2/root/export/home0/hm/samples/
javac Hello.java
Compilation finished at Tue Jan 21 15:39:13
-0:** *compilation* (Compilation:exit [0])--L1--C0--
```

To run the program, select "Shell Command" from the "Tools" menu, and type "java Hello":



```
** *compilation* (Compilati
Shell command: java Hello
```

Now the output will appear in Emacs' "minubuffer" (where you type the commands):



```
** *compilation*
Hello World!
```

6 Closing remarks

Also try Eclipse which is installed on the machines in LinuxLab (room 2A50) and is included in the Fedora Core 4 Linux distribution.

References

- [1] *The j2sdk documentation.* <http://java.sun.com/j2se/1.5.0/docs/index.html>
- [2] *BlueJ Homepage.* <http://www.bluej.org/>
- [3] *Eclipse Home Page* <http://eclipse.org/>