Installing the Code

Detailed instructions for installing, configuring and testing the source code on OS X.

These instructions also work with the free demo version of the guide.

1. Create a folder called '**TIJ4-Solutions**' in your home folder.
2. Copy the zip file containing the code that you received, '**TIJ4-Solutions-code**', when you purchased the guide to the folder '**TIJ-Solutions**'. Unzip the file using the Archive Utility. When you're done, you should see a sub-folder '**TIJ4-Solutions-Code'** and numerous subfolders, including subdirectories corresponding to the chapters in the solution guide.
3. Create a folder called '**jars**' in your recently created folder '**TIJ4-Solutions**'. Place the following files into this directory:
   javassist.jar (download here: http://sourceforge.net/project/showfiles.php?group_id=22866; you may need to search for it).
   swt.jar from the Eclipse SWT library (http://download.eclipse.org/eclipse/downloads/). Click on the most recent build number, then scroll down to "SWT Binary and Source" and select the file corresponding to your platform. Further details about finding the jar file are in *Thinking in Java, 4th Edition*, under the heading "Installing SWT."
   xom-1.1.jar, available from http://www.cafeconleche.org/XOM/.
4. Create a small shell script '**setenv.ksh**' to set the CLASSPATH environment variable in the folder '**TIJ4-Solutions**' using the TextEdit application. Open the application by pressing ⇧⌘P in Finder and selecting '**TextEdit**'. Verify that the text is in plain text format, switching between plain and rich text format can be done by pressing ⇧⌘T. Copy and paste the code below and save the file as setenv.ksh in the folder '**TIJ4-Solutions**' in Unicode (UTF-8) encoding.

```
#!/bin/ksh

if [[ $0 != '-bash' ]]
then
        echo "Execute this script using \". $0\""
fi

CLASSPATH=.:..
CLASSPATH=$CLASSPATH:~/TIJ4-Solutions/TIJ4-Solutions-code
CLASSPATH=$CLASSPATH:~/TIJ4-Solutions/jars/javaws.jar
CLASSPATH=$CLASSPATH:~/TIJ4-Solutions/jars/javassist.jar
CLASSPATH=$CLASSPATH:~/TIJ4-Solutions/jars/swt.jar
CLASSPATH=$CLASSPATH:~/TIJ4-Solutions/jars/xom-1.1.jar

export CLASSPATH
```

5. Download the Ant 1.7 (or newer) build tool at http://ant.apache.org/. Start a terminal by pressing ⇧⌘U in Finder and selecting 'Terminal'. Type the following instructions at the command prompt. After entering the 'sudo' instruction a password is requested, enter the password here you used to logon to your mac.


```
sudo sh
cd /usr/local
mv ~/Downloads/apache-ant-1.7.1 /usr/local
chown `who am i | awk '{ print $1 }'` apache-ant-1.7.1/
ln -s apache-ant-1.7.1 ant

cd /etc
chmod u+w bashrc
echo "export ANT_HOME=/usr/local/ant" >> bashrc
echo "export PATH=\${PATH}:\${ANT_HOME}/bin" >> bashrc
chmod u-w bashrc
exit
exit
```


6. Test Ant by by opening new terminal by pressing ⌘N and typing 'ant '. This should print 'Buildfile: build.xml does not exist! Build failed' .

   Note: **Ant** is required in order to compile the examples in the book. Once you successfully run '**ant build**' in the root directory, you can also compile each example individually (once you have the CLASSPATH set, as described in Step 4) using the **javac** command-line compiler. To compile a file called **MyProgram.java**, you type **javac MyProgram.java**.

7. Once you've run ant build in the root directory, you can also move into individual chapters and type ant (to compile and execute the code in that chapter) or ant build (to compile the code only).

8. This code is designed to work without an IDE, but it has also been tested with Eclipse (free at http://www.eclipse.org/); see the following section for instructions on how to use the code with Eclipse.
   If you want to use this code inside other IDEs you might need to make appropriate adjustments. Different IDEs have different requirements and you might find it's more trouble than it's worth right now; instead, you may want to begin with a more basic editor like JEdit (free at http://www.jedit.org/).

9. Note: The output for the programs has been verified for Java 6. Certain programs (primarily those that use hashing) can produce different output from one version to the next.