Inf2-SEPP 2023-24

Development tools

1 Introduction

This document describes the required tools that you will need to use for your Inf2-SEPP Coursework 3, as well as details for installing and configuring them on your local machines, and accessing them remotely from DICE (to be done only if you have no other alternative, see below).

2 The Required Tools for Inf2-SEPP

All of the software required for this course is pre-installed on the Informatics DICE[1] systems. However, we would strongly recommend that you prioritise installing the software locally on your machine, and you only use remote access to DICE as a fallback in case you have problems with the software on your system. This is because it is likely you will find using DICE remotely unacceptably slow.

2.1 Java

Inf2-SEPP uses the Java programming language, because of its excellent tool support, focus on an object-oriented approach and popularity in the industry. However, the course does not teach Java but rather relies on your experience with it from Inf1-OOP. Apart from the resources in this first year course, we have also provided some resources on Java under the Learn page of the course- Resources.

We use OpenJDK[2] which is a freely available implementation of the Java platform. You can use Oracle JDK too. Whatever you choose, your code must be compatible **JDK version 14**. You can check your installed Java version as follows:

```
$ java --version
openjdk 14.0.2 2020-07-14
OpenJDK Runtime Environment (build 14.0.2+12-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 14.0.2+12-Ubuntu-120.04, mixed mode, sharing)
```

[1] http://computing.help.inf.ed.ac.uk/what-is-dice

[2] https://openjdk.java.net/

Note that we are using '$' to represent the terminal prompt (a message that usually shows which directory you are currently in); this is not something you are meant to type.

If you have JDK 14, then you can just use it as-is. But if you have a newer version on your machine, you will need to be very careful to ensure that you do not rely on new Java features introduced after JDK 14. This is to allow markers to run your code without requiring them to match all possible students' setups. So, for example, Record classes are not allowed, as they were only introduced in Java 17.

Section 5 shows how to lower your active JDK version to 14 in IntelliJ, provided that you have a higher version installed.

2.2 Maven

We use Maven for dependency management in the assignment. You do not need to install anything, it comes bundled with IntelliJ.

2.3 IntelliJ IDEA

It is quite possible to create Java programs with a simple editor and compile and run them from the command line (possibly using a standalone install of Maven). The alternative is to use an IDE (integrated development environment), which provides a graphical interface to an editor and file browser with specific support for code development.

Building on Inf1B, we have decided to use IntelliJ IDEA and we strongly recommend that you use this - it is a good environment with support for Maven, and very popular amongst Java developers.

The "Community Edition" of IntelliJ is freely available from the JetBrains[3] website for you to install on your own machine. We recommend that you accept all of the default values when this is initially started – these can be changed later if necessary.

2.4 Version Control

You should use a Version Control system to manage your coursework project. But it is up to you which one you use. The most common and best support option is git, but you could also use Mercurial (hg) or Subversion (svn). See `https://opencourse.inf.ed.ac.uk/inf2-sepp/other-resources/other-tools` for guidance about them.

[3] `https://www.jetbrains.com/idea/`

3 Installation and Configuration

The sections below provide advice on installing and configuring the required tools for this course. We will also be providing help for this during the labs in Weeks 8-9 of the course, and by using Piazza.

Important notes:

- We recommend that you install the software on your own machines, but also *make sure that you can run your software on some alternative system (e.g. DICE) as well, in case your own machine has a problem.* Failure of your own machine will not be taken into account when assessing your assignment submissions.
- *Make sure that all your assignment code works with JDK 14 as described earlier.* The recommended version has been carefully chosen for consistency with taught features and available software on university machines. If your code does not compile and run with JDK 14, you will lose marks.

3.1 Installing the Tools on Windows

Windows x64 zipped binaries for OpenJDK are available from the OpenJDK website[4]. Unzip these into a directory without spaces in the path[5]. You will then need to add these directories to the system PATH variable[6], for example, you could place the JDK binaries into `C:\dev\jdk-14.0.2` and add `C:\dev\jdk-14.0.2\bin` to PATH.

You can check that this works correctly by checking the installed Java version (as described above) in a (new) PowerShell window.

IntelliJ IDEA has a Windows version available on its websites which can be installed in any directory.

[4] `https://jdk.java.net/archive/`

[5] If you have spaces in the directory path, the path will need quoting when you use the command line tools.

[6] see `https://www.java.com/en/download/help/path.html`

3.2 Installing the Tools on MacOS (Catalina)

The easiest way to install OpenJDK and Gradle on the Mac is to use the Brew package manager. If you don't already have brew installed, copy the command on the Brew website[7] and paste it into a terminal. You should then[8] run the following commands:

```
$ brew tap
$ brew homebrew/cask
$ brew homebrew/cask-version
```

You can now install OpenJDK:

```
$ brew install java
```

You will probably want to set some environment variables to make these available by default:

```
$ echo 'export PATH="/usr/local/opt/openjdk/bin:$PATH"' >> ~/.zshrc
$ echo 'export CPPFLAGS="-I/usr/local/opt/openjdk/include"' >> ~/.zshrc
```

IntelliJ IDEA has a Mac package available on its website.

3.3 Installing the Tools on Linux

The process for installing the tools on Linux will depend on the specific version of Linux that you are using.

On Ubuntu and related distributions, the appropriate version of OpenJDK can be installed with apt:

```
$ sudo apt-get update && sudo apt-get upgrade
$ sudo apt install openjdk-14-jdk
```

Intellij IDEA can be installed relatively painlessly with Snappy:

```
$ sudo snap install intellij-idea-community --classic
```

[7] https://brew.sh/

[8] https://gist.github.com/gwpantazes/50810d5635fc2e053ad117b39b597a14

4 Using the Tools on DICE

All of the necessary software is installed on the Informatics DICE systems which can be accessed remotely via the XRDP Service[9]. Again, we recommend that you install the tools on your local machine and only use DICE as a fallback as it will likely be very slow.

No special configuration is required on the current version of DICE, and all of the tools can be launched from the command line: `java ...`, `mvn ...`.

IntelliJ IDEA interacts badly with the AFS filesystem[10] used in DICE. This can cause the Open Project dialog to hang. You can work around this by opening IntelliJ directly in the project directory:

```
$ cd path/to/your/project/
$ ideaIC .
```

The "." on the second line is very important!

5 Limiting JDK features to v14 in IntelliJ

If you have a higher version of JDK than the required 14, there is a way to limit the functionality of your project in IntelliJ to that of version 14. See the screenshots and their captions provided on the next page.

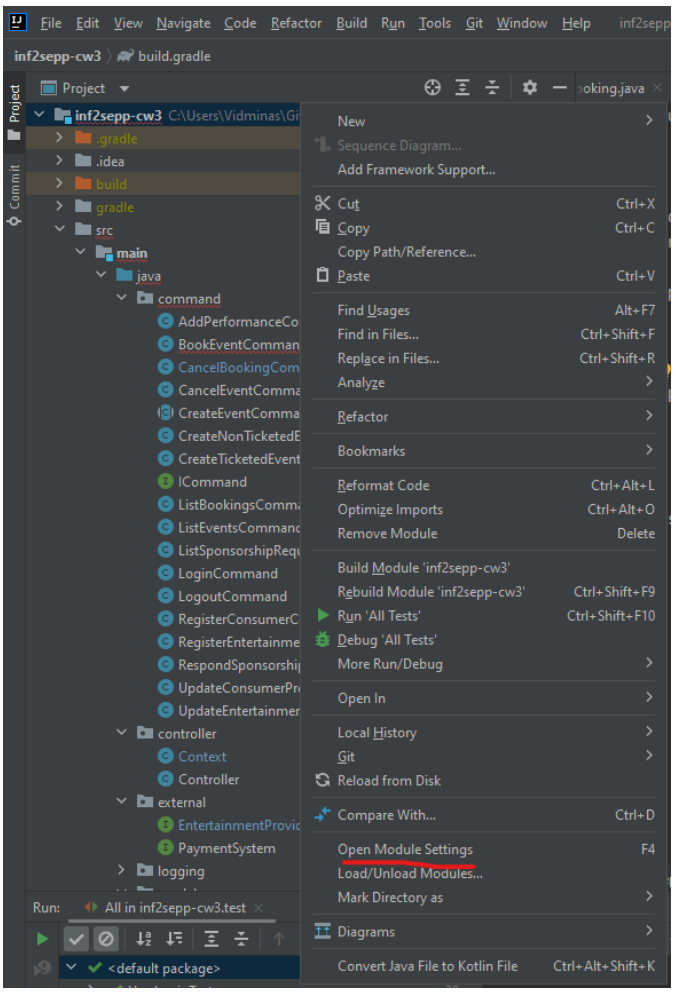
This generally applies to projects in IntelliJ. But when using a Maven project, the IntelliJ language level setting gets overwritten by the configuration in `pom.xml`. Your `pom.xml` should contain lines like:

```
<properties>
    <maven.compiler.source>14</maven.compiler.source>
    <maven.compiler.target>14</maven.compiler.target>
</properties>
```

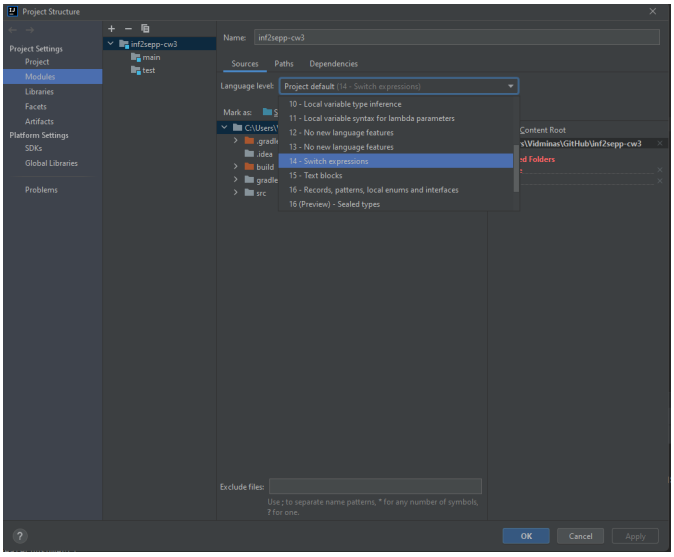
Make sure the source and target level are both set to 14 like above.

[9] `http://computing.help.inf.ed.ac.uk/remote-desktop`

[10] see `https://intellij-support.jetbrains.com/hc/en-us/community/posts/206924155-Problems-with-afs-access`



Step 1: Right-click your project, and select "Open Module Settings."



Step 2: From the "Language Level" drop-down menu, select 14.