Getting Started with IntelliJ

COMSM0086

Dr Simon Lock

Overview

- Downloading and installing IntelliJ IDE
- Dealing with User Agreement and Licensing
- Open an existing template project
- Install the Java Development Kit (JDK)
  (including compiler, runtime and libraries)
- Running your first Java program !

Download from JetBrains

Make sure you get the download for your platform !

(Website _should_ autodetect to the right option)

Windows macOS Linux

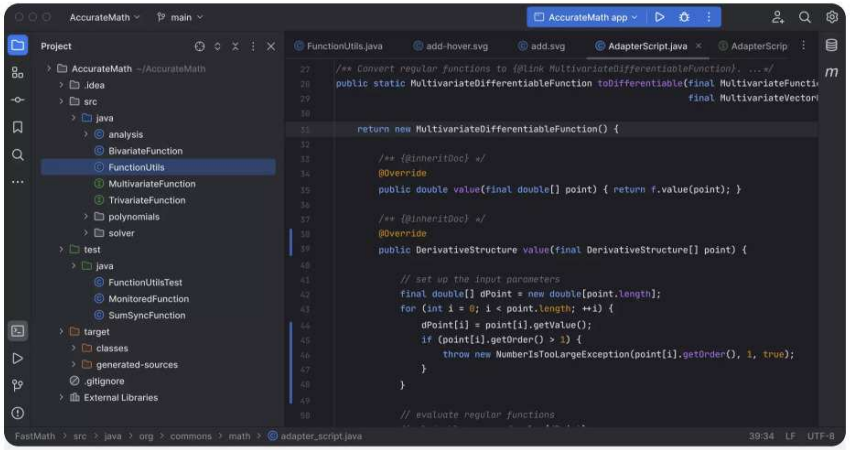
IntelliJ IDEA Ultimate

The Leading Java and Kotlin IDE

Download .dmg

Free 30-day trial

Select an installer for Intel or Apple Silicon



Version: 2023.3.2
Build: 233.13135.103
20 December 2023

System requirements

Installation instructions

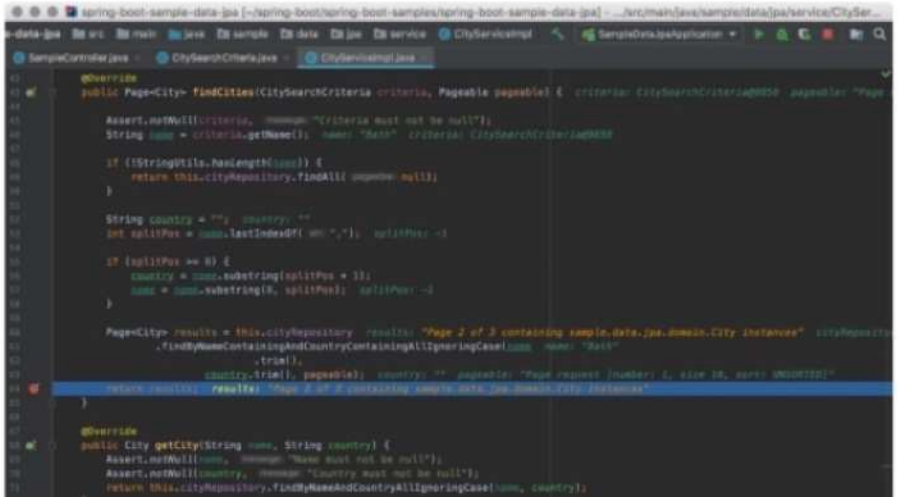
Other versions

Third-party software

Open the download and install as normal !

Alternative Approach

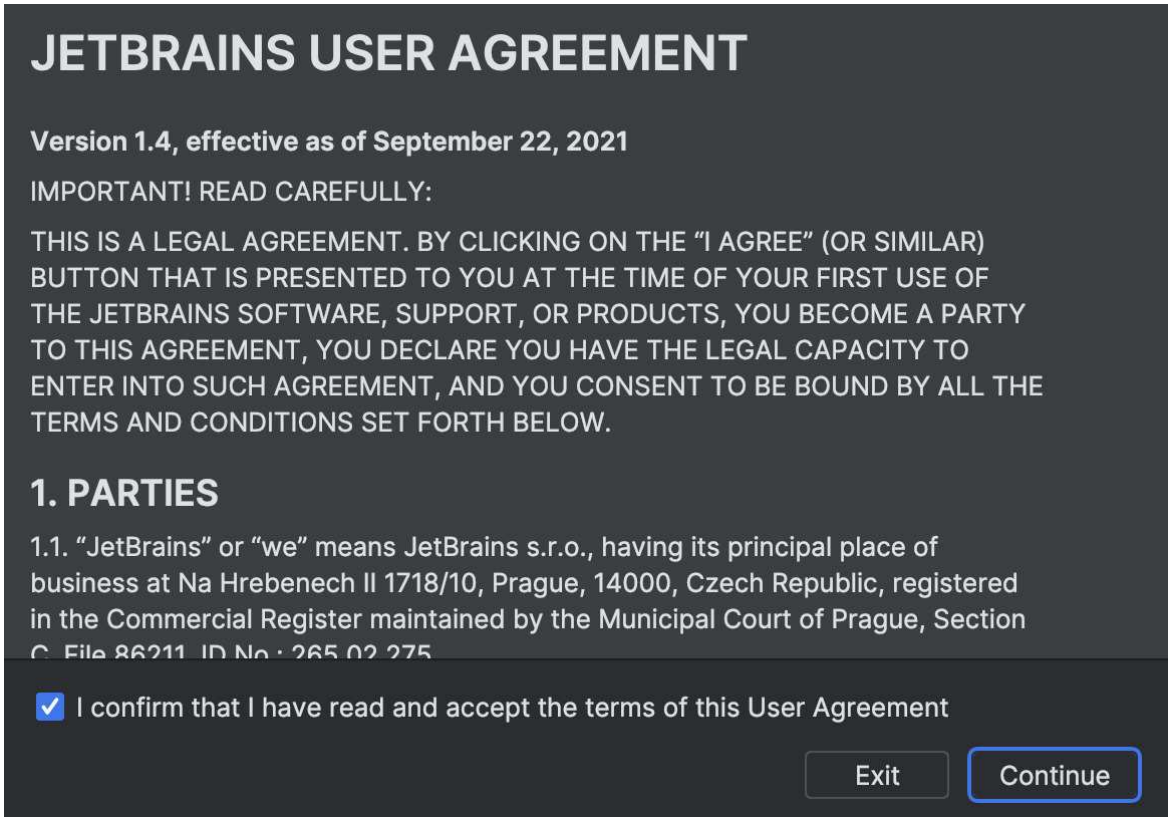
IntelliJ might be available through your platform's Package Manager (if it has one !)



User Agreement

First time you run IntelliJ, you'll see User Agreement

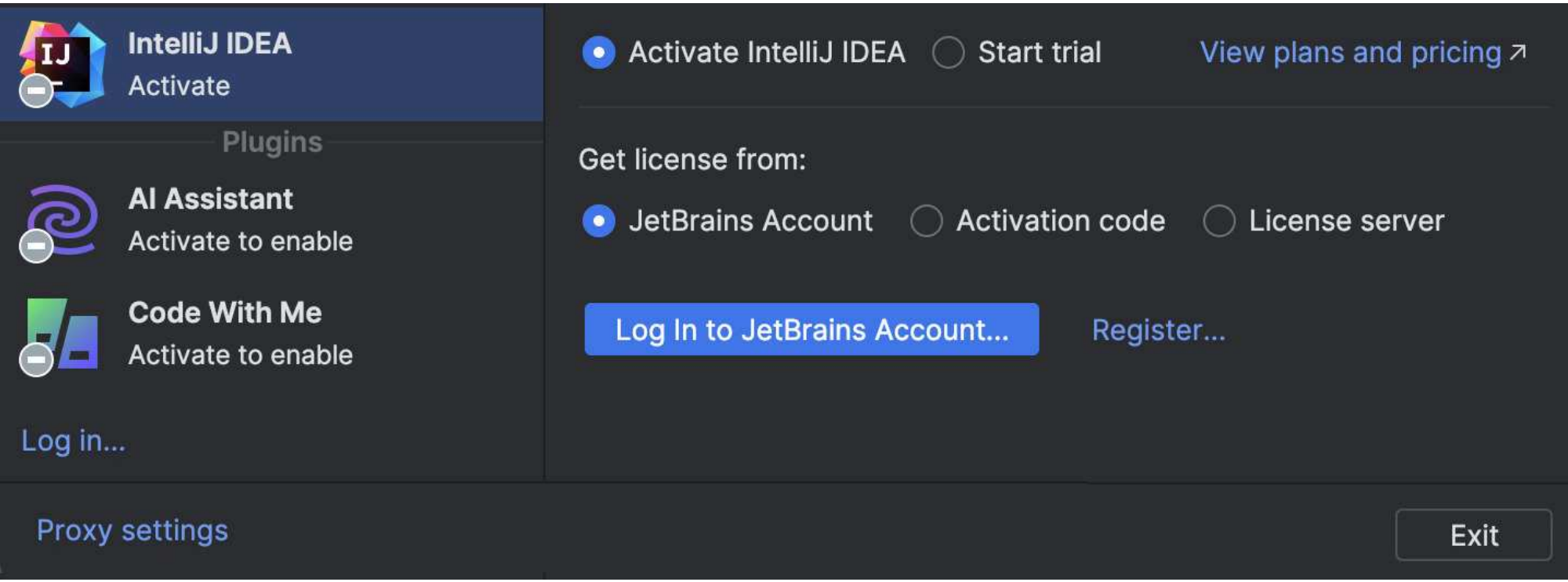
Tick the box and click continue if you are happy !



License Settings

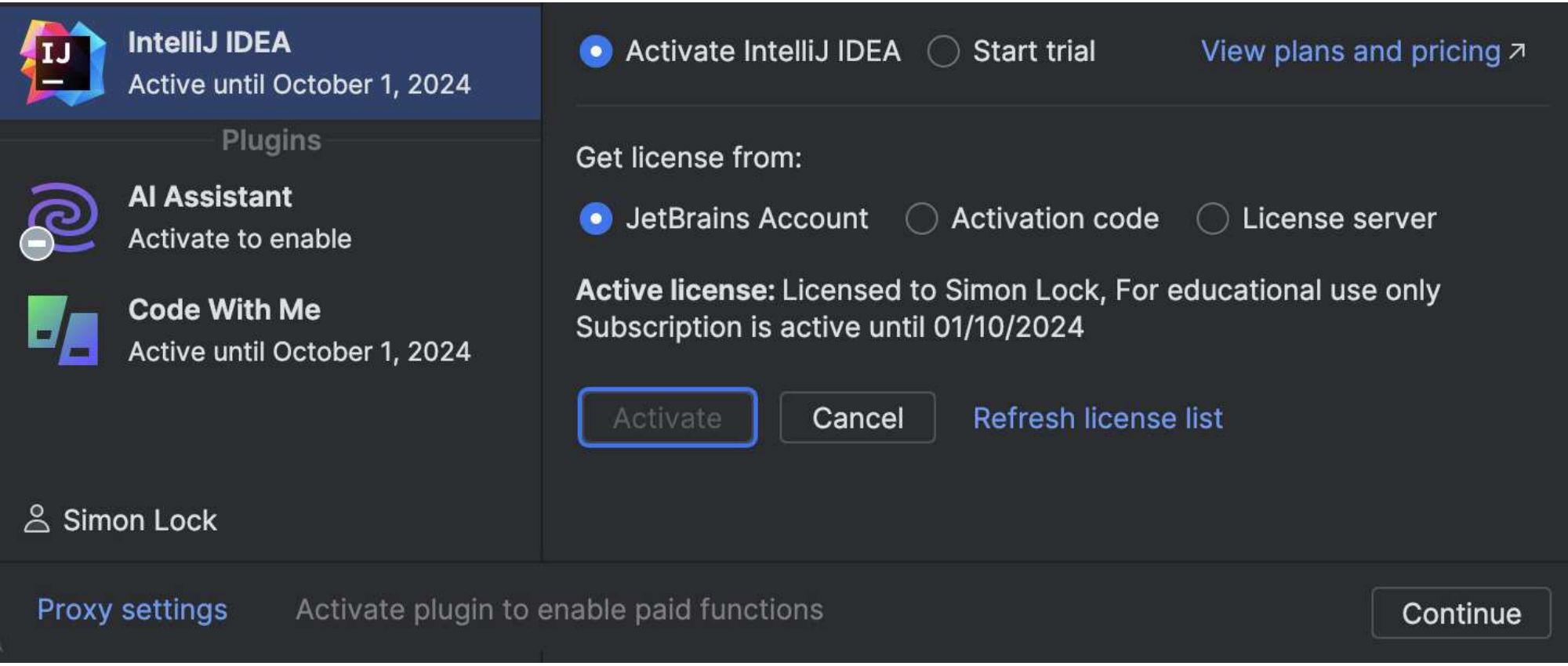
"Start Trial" if you haven't registered with JetBrains

"Log In to JetBrains" if you have educational license



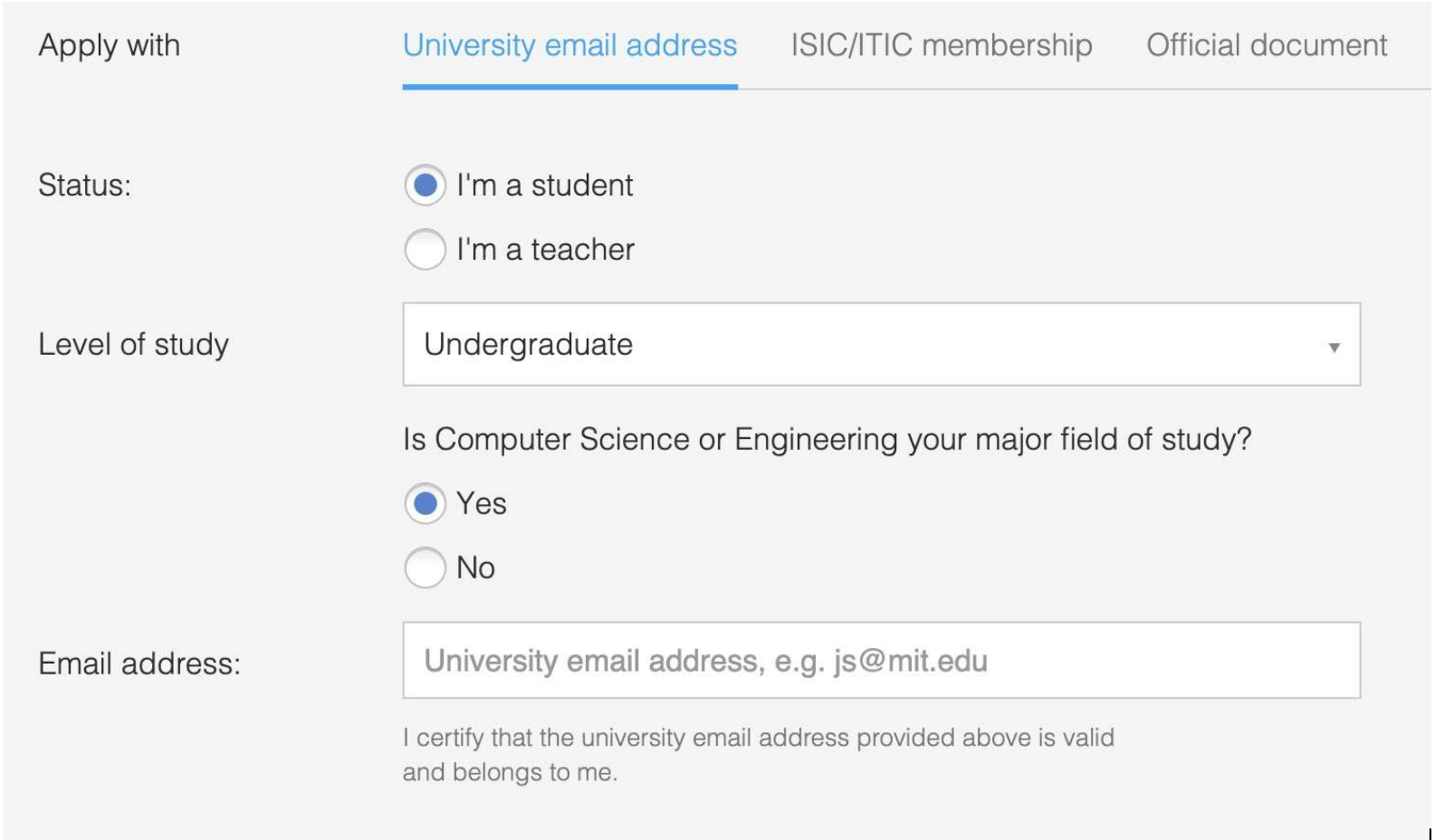
Extra Step

Make sure you click the "Activate" button !



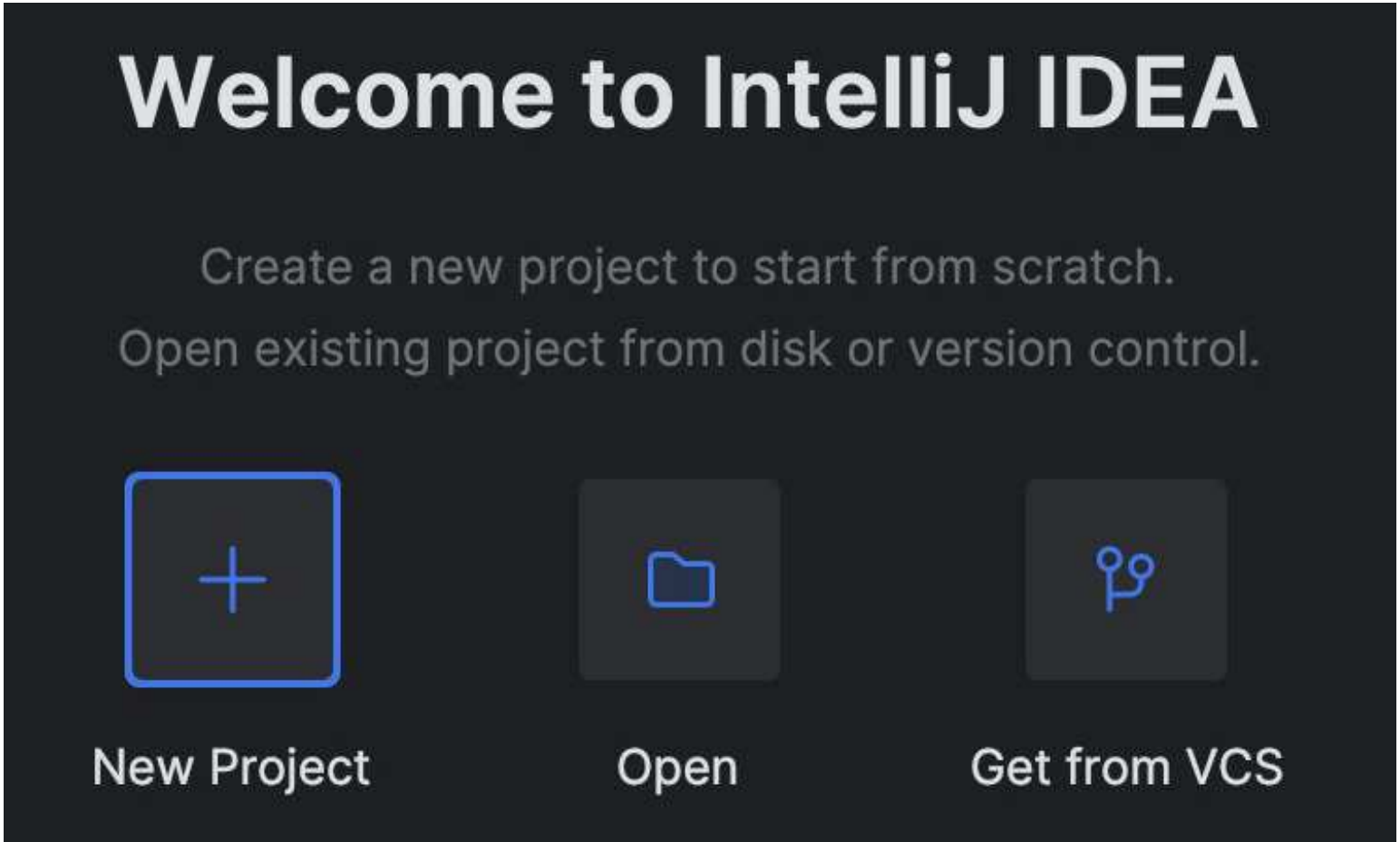
Register for an Educational License

https://jetbrains.com/community/education/#students



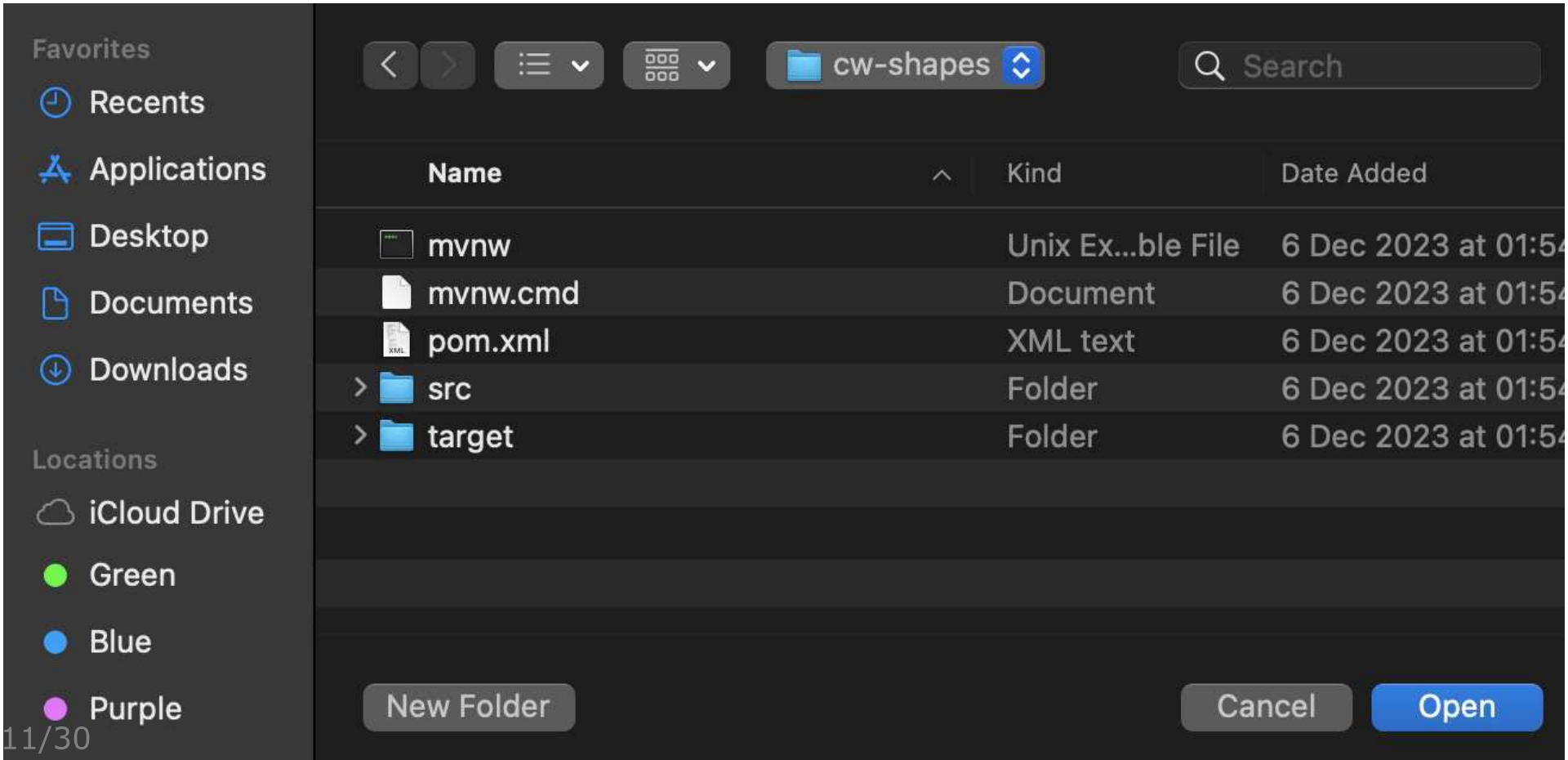
Welcome Screen

Most of the time on this unit you'll need "Open"



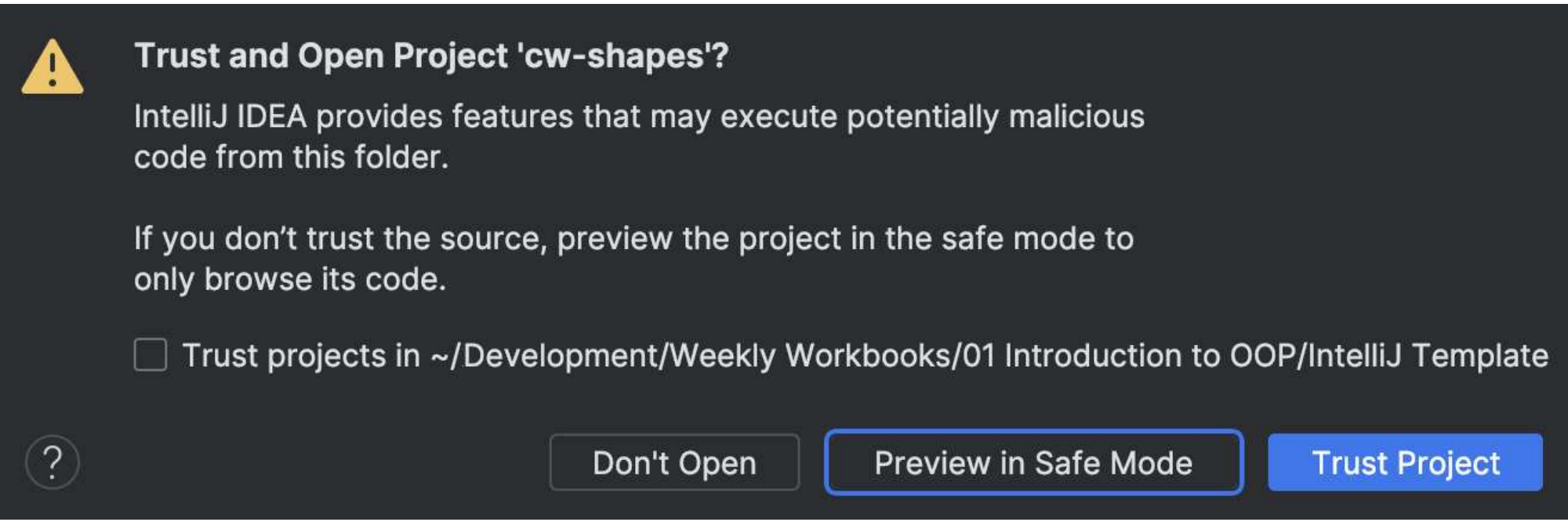
Project Template

Find and open folder containing the pom.xml file



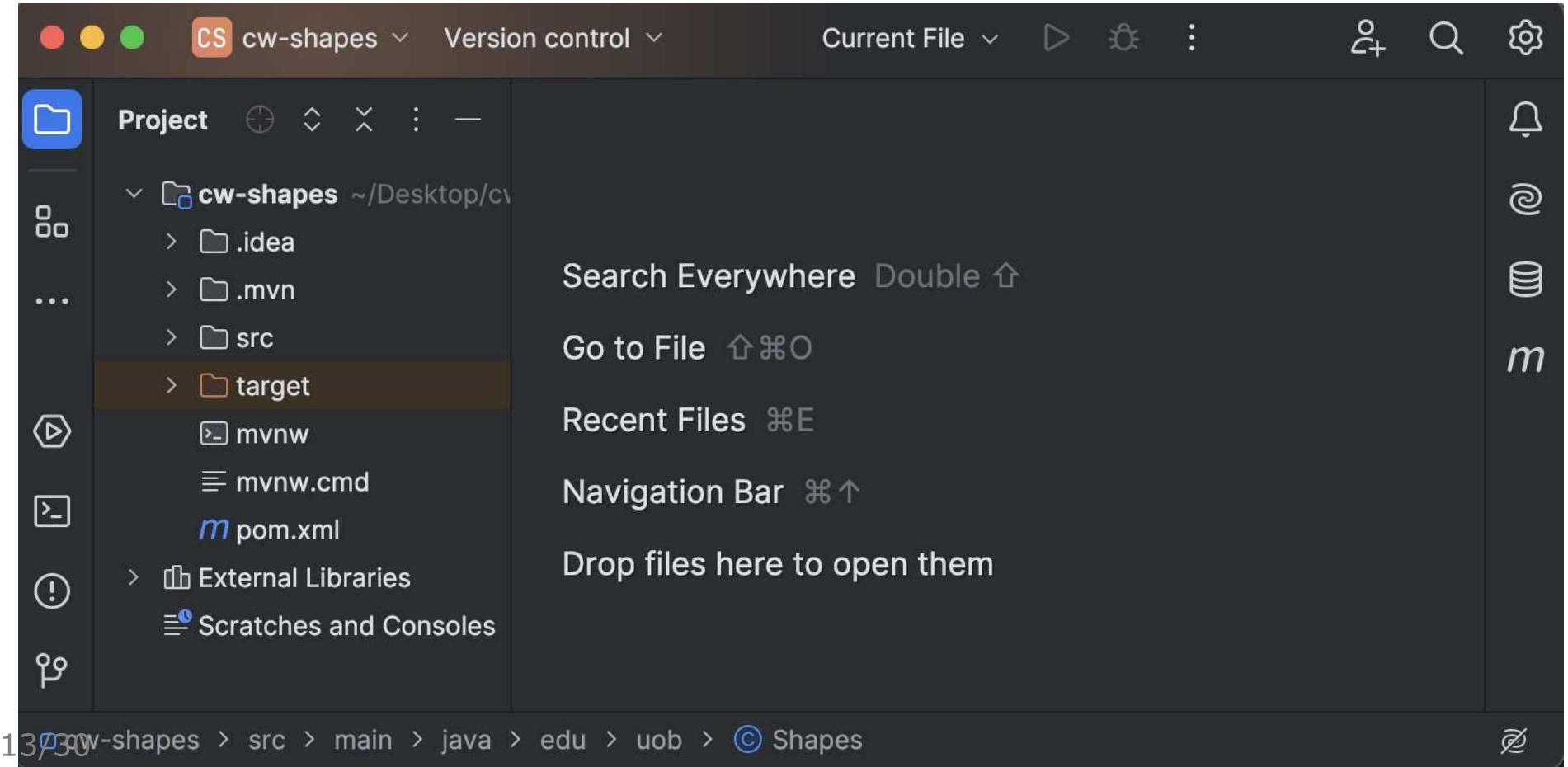
Do You Trust Us ?

Be careful with project from other sources !



Project Opened

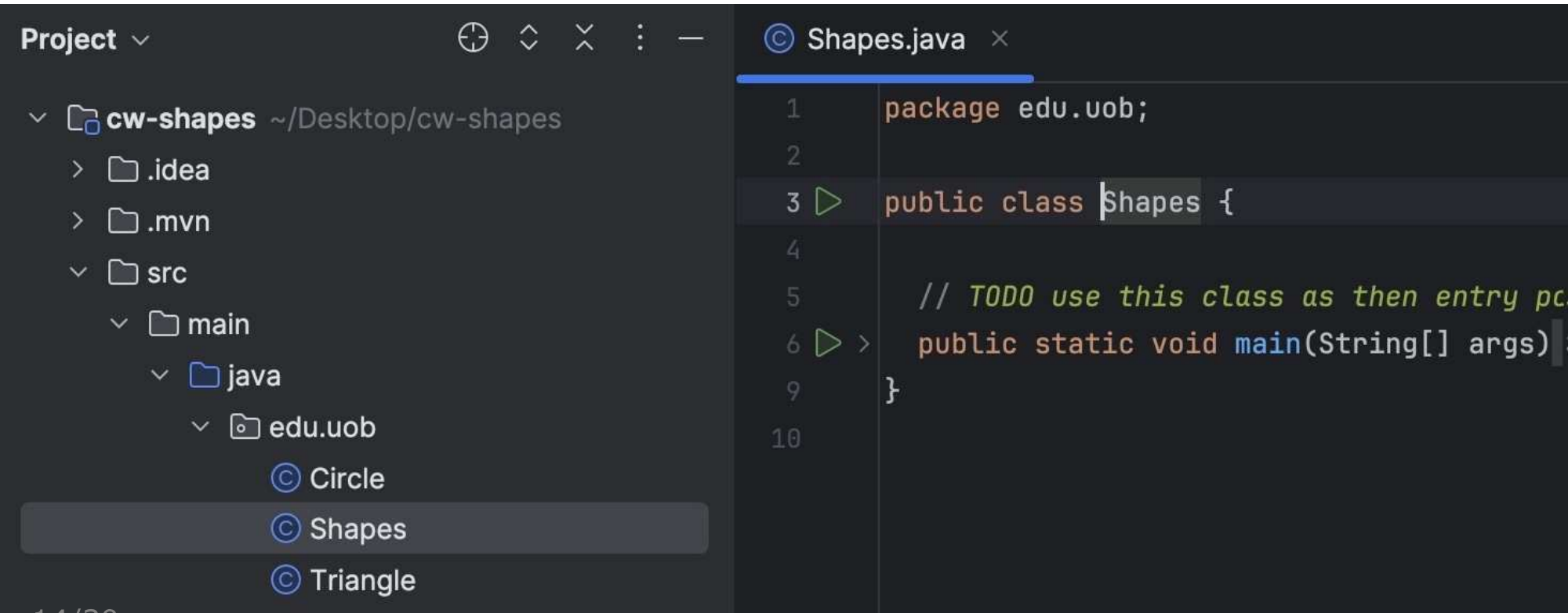
A successfully opened project will look like this:



Open the Main Class

Let's explore the project view to find the main class

In this project, the main class is a file called "Shapes"

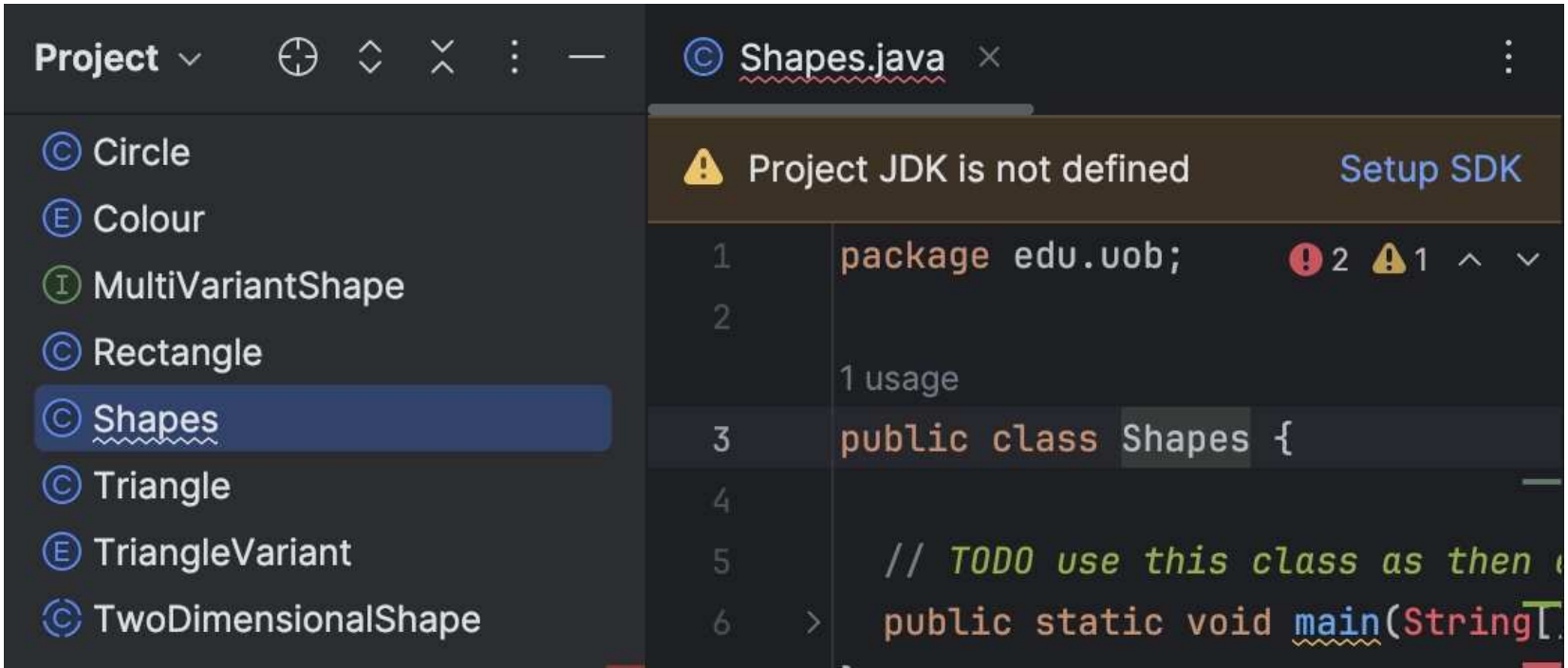


Project JDK is not defined !

IntelliJ is just an IDE - is has no built-in compiler !

We need to install a Java Development Kit separately

This includes compiler, runtime and various libraries

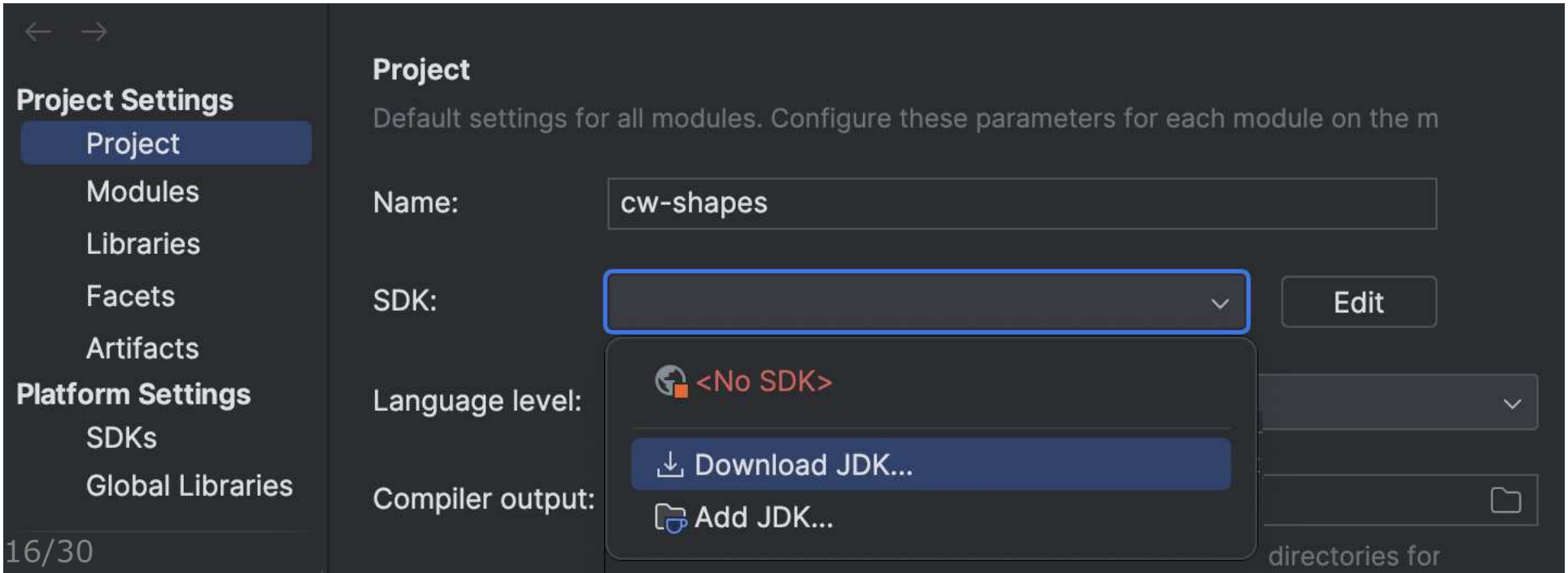


Project Settings

Need to select a JDK to use to compile & run project

Could use an existing JDK (if you have one installed)

Or download a new one from list of those available...



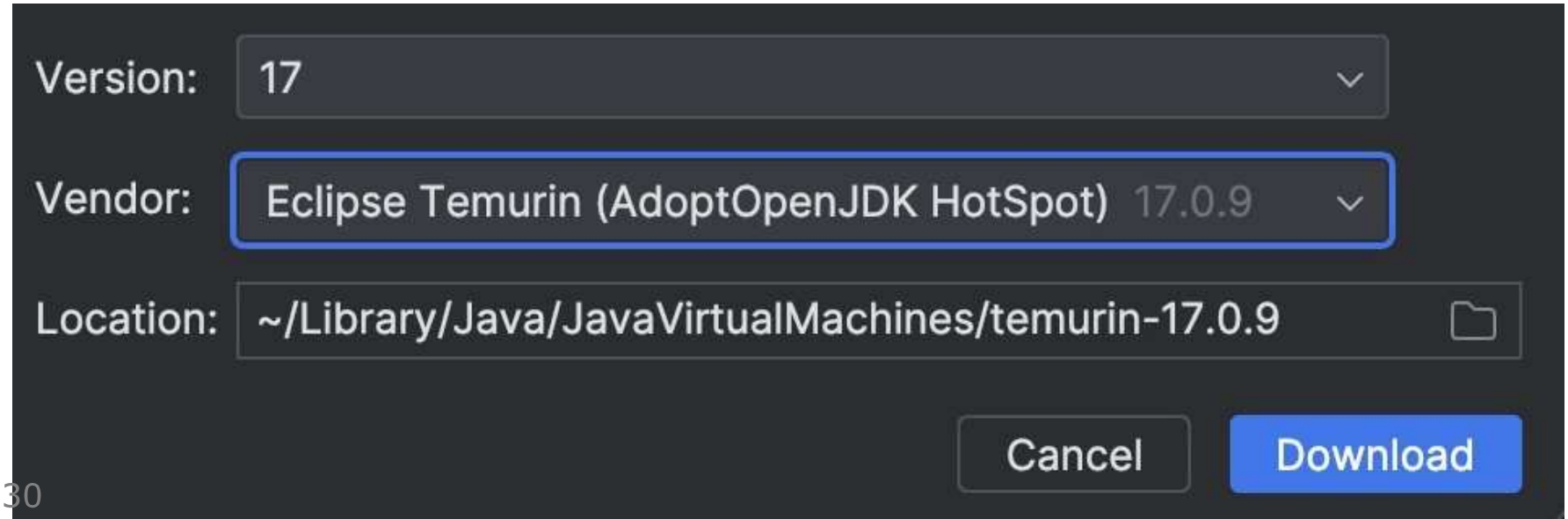
Download JDK

Select JDK to automatically download & install

Lowest Common Denominator: Lab has Java 17

(That's where we are going to mark your code !)

You should choose: Eclipse Temurin (AdoptOpenJDK)



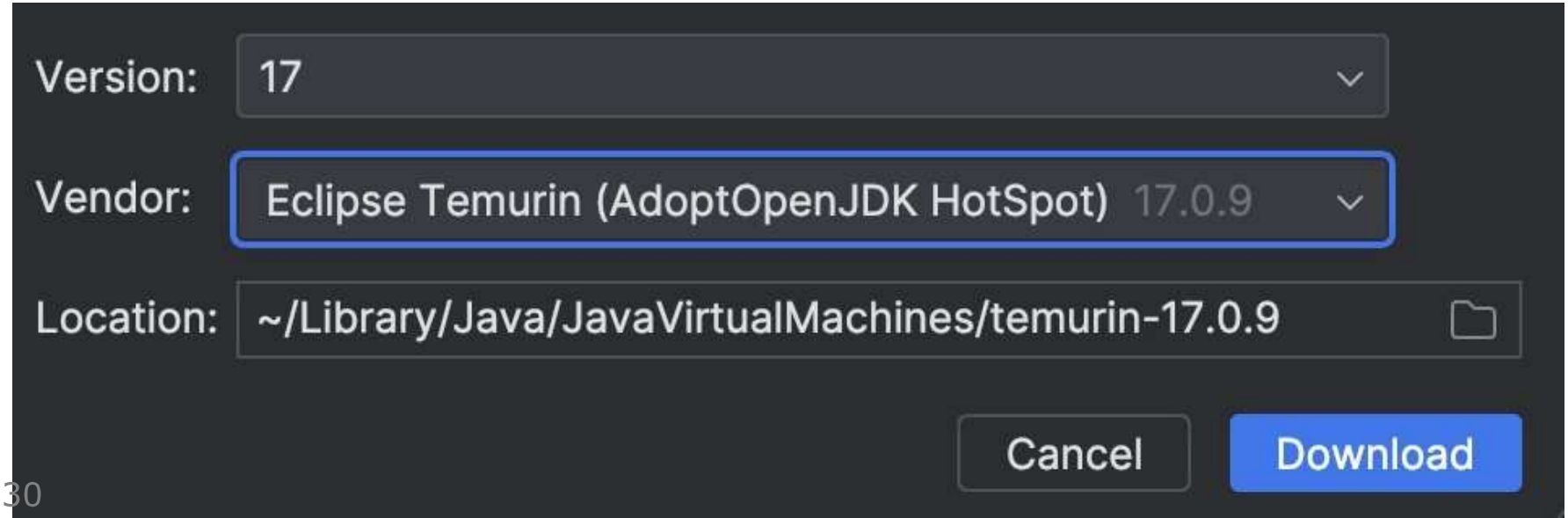
Installation Location

Keep a note of where IntelliJ will install the JDK

The location will _probably_ be something like:

`~/Library/Java/JavaVirtualMachines/temurin-17.0.9`

You'll need this later (to compile on command line)



Be Patient !

It takes a while for the JDK to download and install

Don't worry, only happens once (when you first install)

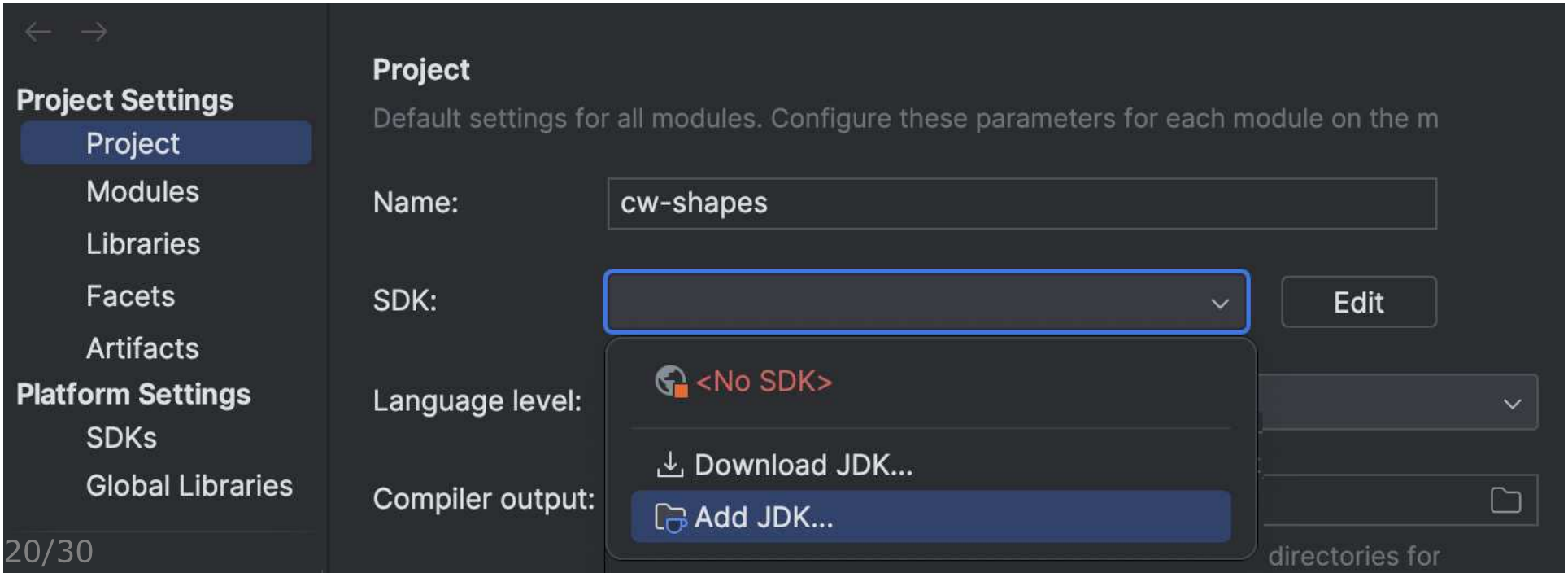
Keep an eye on how things are going on progress bar:

Using Existing JDK

If you already have a JDK installed, you can use that

This will save you having multiple versions installed

It *might* be listed in dropdown, if not click "Add JDK"



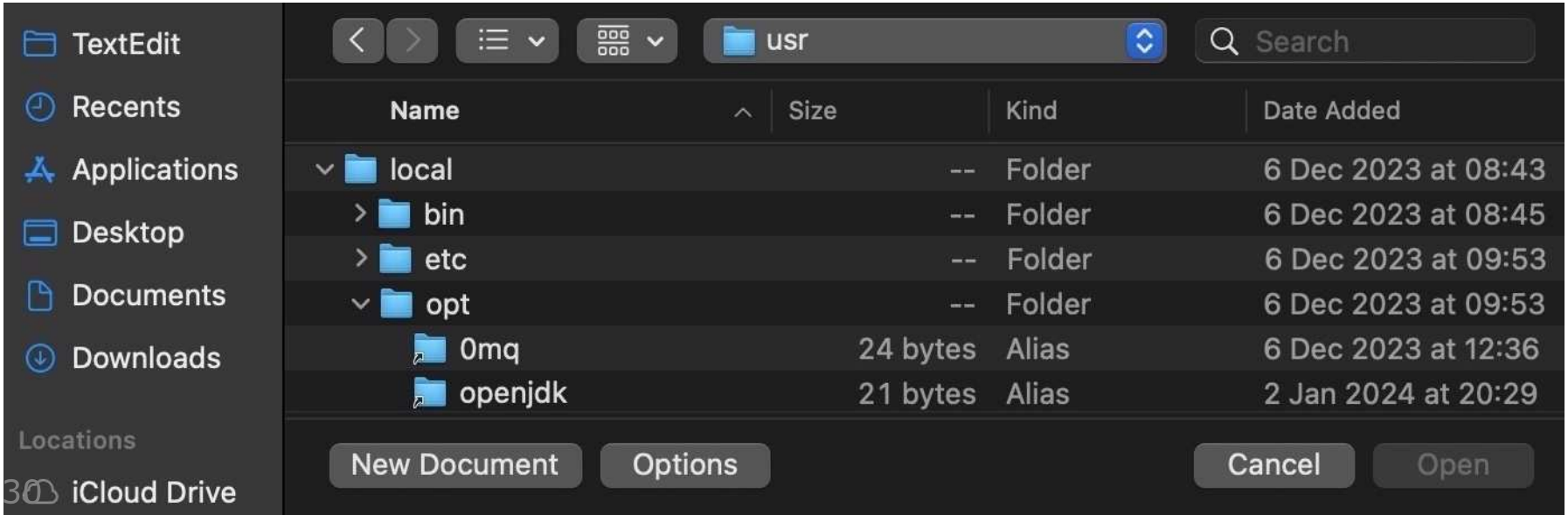
Hunting down Existing JDK

IntelliJ is clever and will usually find installed JDKs

I installed JDK using "Homebrew" package manager

Homebrew is good at hiding installed software ;o)

So I needed to manually hunt around to find the JDK

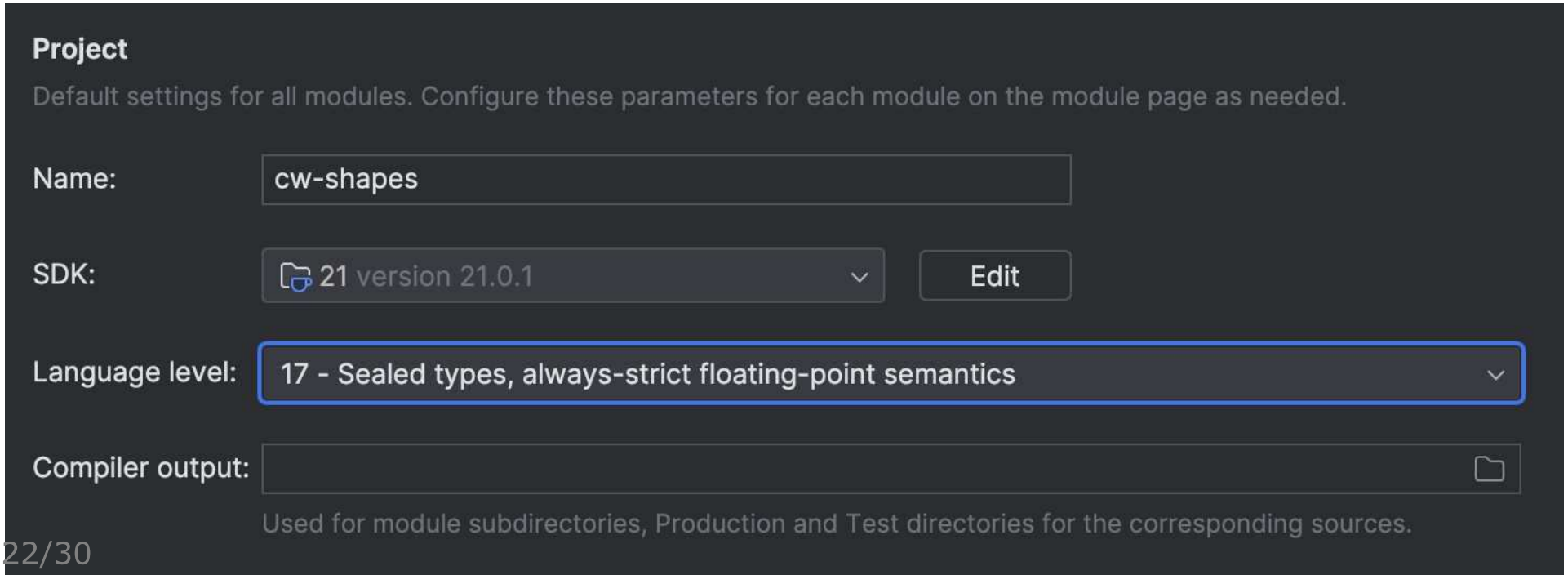


JDK Version

The version I have installed is more recent than 17

However, we should limit language features to 17

To prevent use of new features that won't work in lab

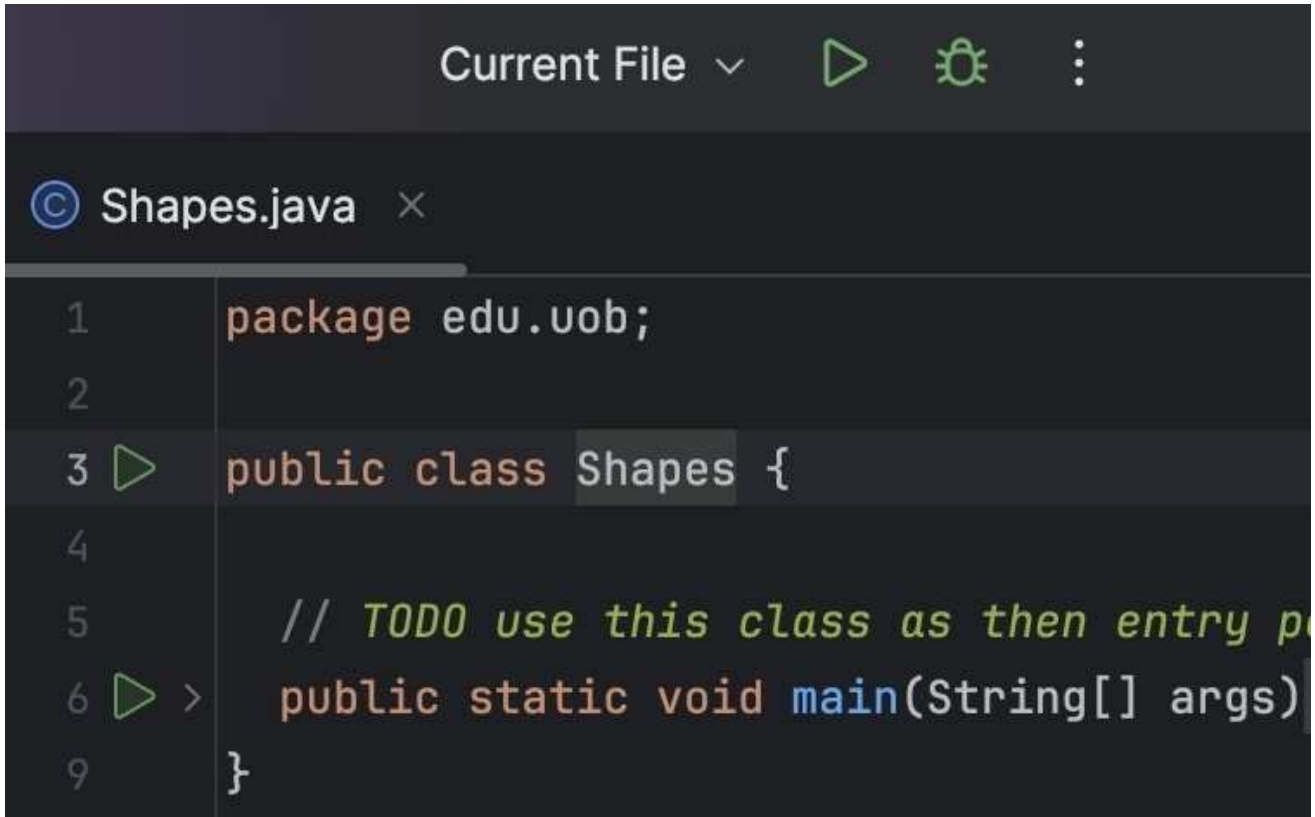


Back to the Project

Ready to Run !

With the main class open in the editing panel

You should now see the green "run" button at the top



Success !

It might take a while for IntelliJ to run the code

It has to build a lot of files the first time around

If everything worked OK, you should see...

Command Line

Although we will be using IntelliJ most of the time

It is useful to also be able to use the command line

Coursework will be marked on the command line

It's essential that you check your code runs there !

In order to be able to compile and run your code

You must tell command line where to find the JDK...

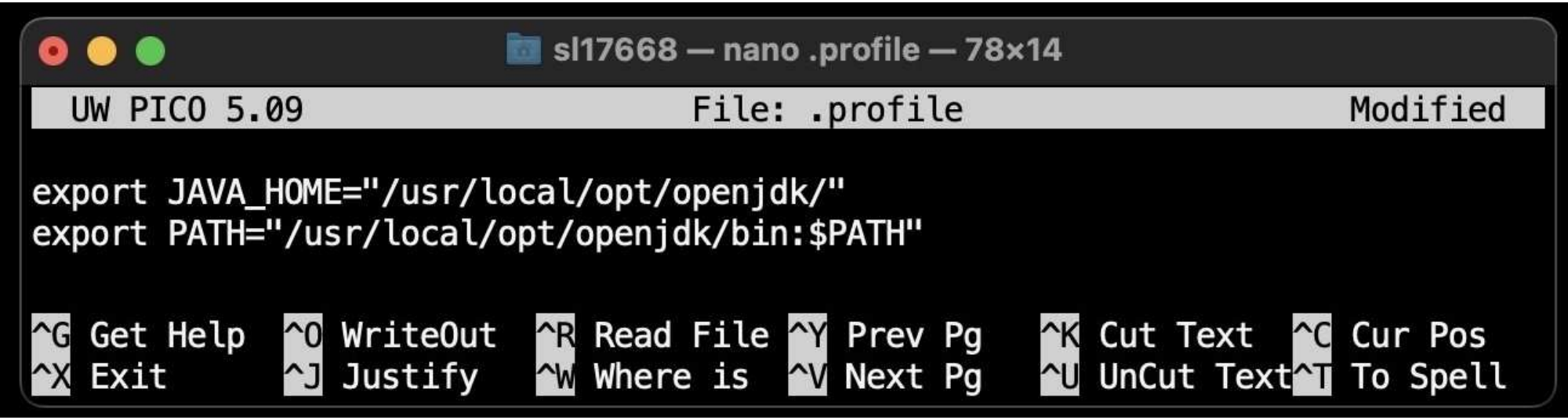
Environment Variables

Add two environment variables to your .profile file

(this is found in your user home folder ~)

JAVA_HOME must point to your installed JDK folder

You must also prepend the JDK *bin* folder to $PATH



Note:

`/usr/local/opt/openjdk/`

Is the location of MY installation of JDK

YOURS is going to be somewhere different !

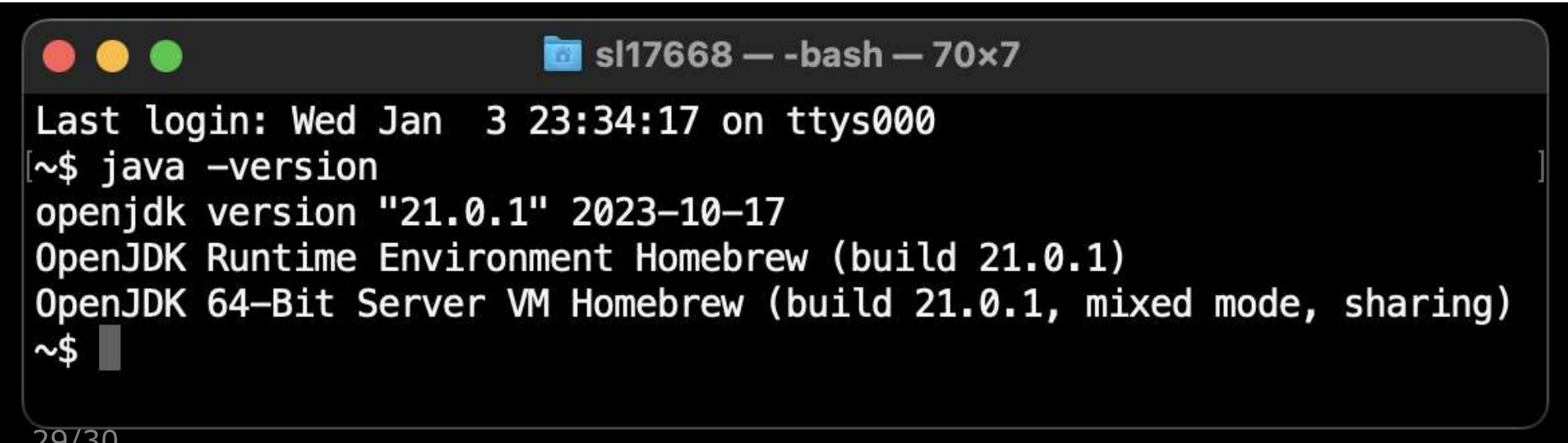
(wherever IntelliJ said it was going to put it)

Testing Your Environment

Open up a fresh terminal window and type:

```
java -version
```

If everything worked, you'll see something like:



Windows

Environment Variables in Windows are set differently

Using a graphical interface in system preferences

See separate guide to setting up JDK on windows