

How to use IntelliJ to create Greenfoot projects

By RcCookie

Introduction

This short tutorial will guide you to create and run your first Greenfoot project right out of IntelliJ IDEA. There are plenty of advantages in doing so:

- You get to use all the features of IntelliJ, especially code highlighting, auto completion and access to documentation of Greenfoot and the Java libraries
- You can take a look into the sources of Greenfoot and see how it is coded itself
- Be more flexible with your programming style, use packages, Java internal classes, and similar
- Make use of advanced programming features, like debugging or using build tools like Maven to use and create 3rd party libraries
- Many more...

Not all of the following steps have to be performed every time; some only have to be done once and can be reused in following projects.

If you are experiencing any issues, feel free to contact me on [our Greenfoot-Discord-Server](#).

Side note: the instructions are intended for Greenfoot 3.6.0 and later, ever since Greenfoot switched from Java 8 to Java 11. For older versions some steps regarding JavaFX can be skipped, but I have not ever tested this myself. Also things may work differently on MacOS or Linux.

Step 0 - Installing the necessary programs

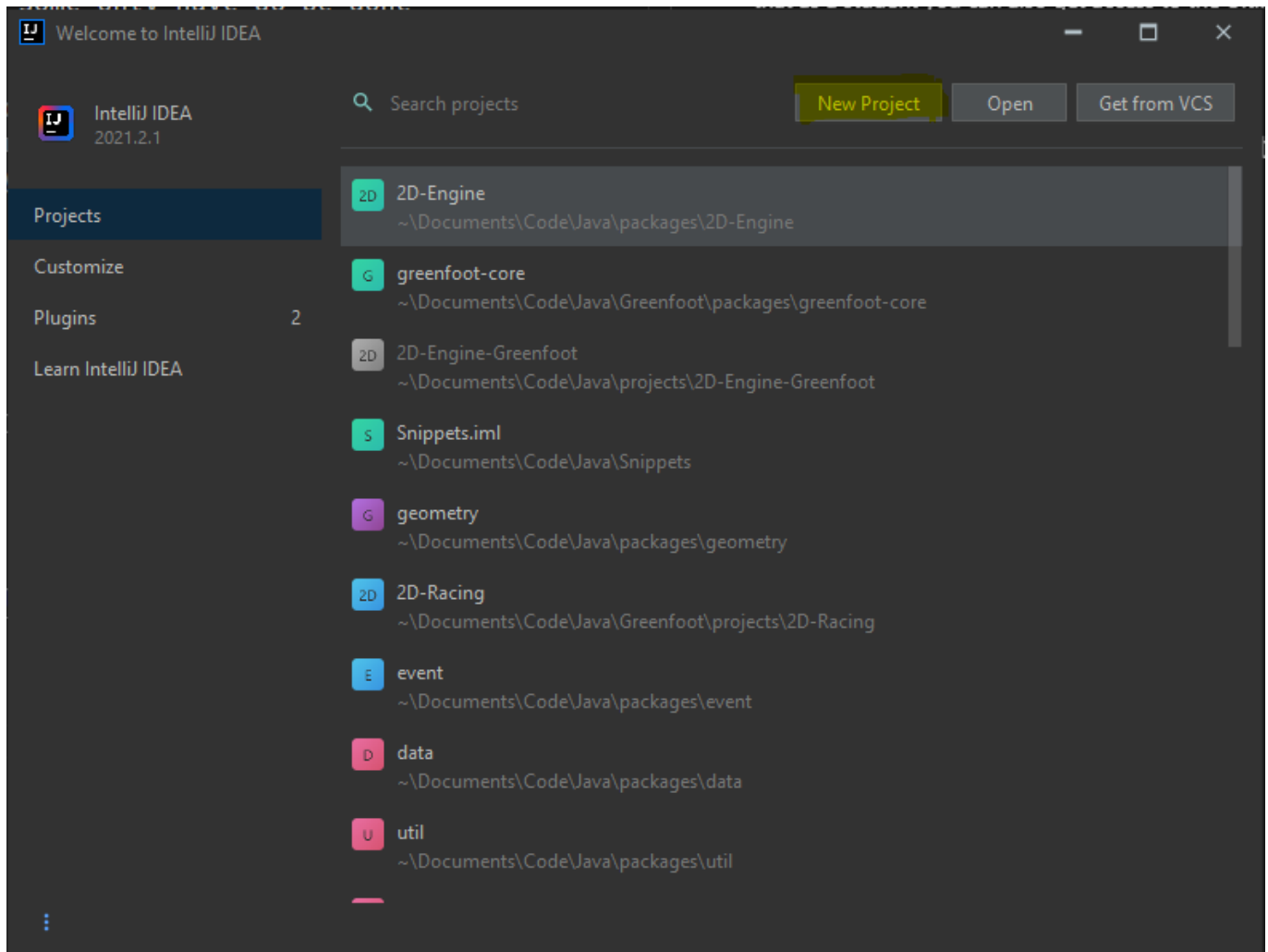
You will need to install both Greenfoot and IntelliJ IDEA. For IntelliJ, both the Community and the Ultimate version will work, but Community is free. Note though that as a student you can also get access to the Ultimate version for free. For Greenfoot, you also need to download the source code. The Greenfoot sources are available on the Greenfoot website, but it's quite complicated to order the files in the folder you get, and also some files are missing. I've created a sorted collection of all the files you need. The sources in that are for Greenfoot versions 3.6.X and 3.7.X, if you use a different version you will *also* have to get the corresponding sources from the Greenfoot website (you still also need my files).

- [Greenfoot application](#)
- [Combined Greenfoot sources](#)
- [Greenfoot source code \(only non 3.6.X and 3.7.X versions\)](#)
- [IntelliJ IDEA](#)

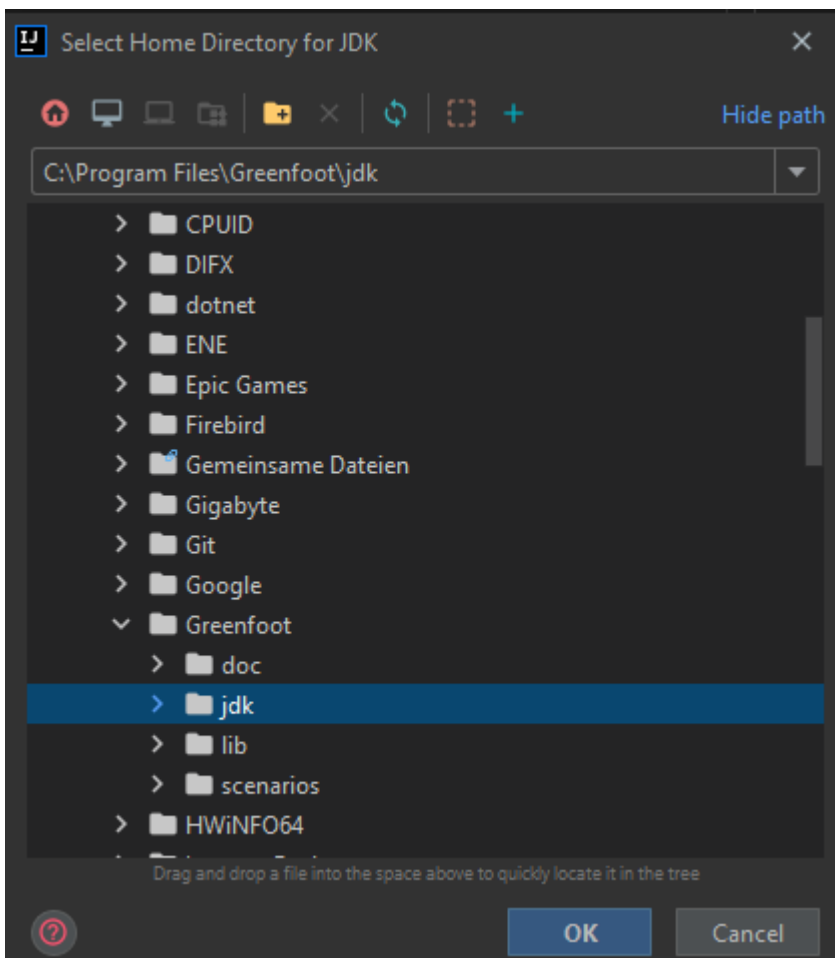
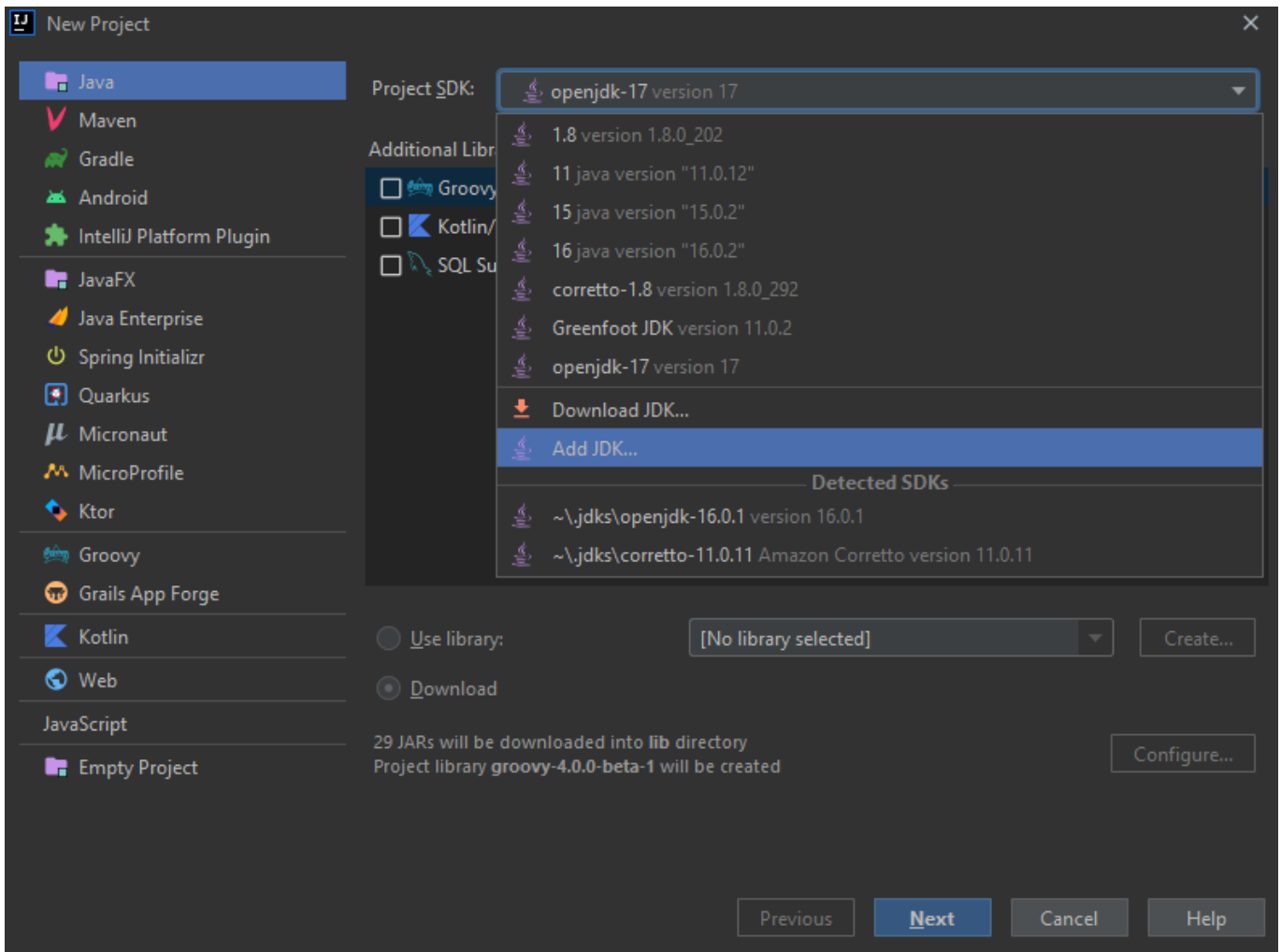
Extract the combined sources into some folder you can remember.

Step 1 - Create a new project in IntelliJ

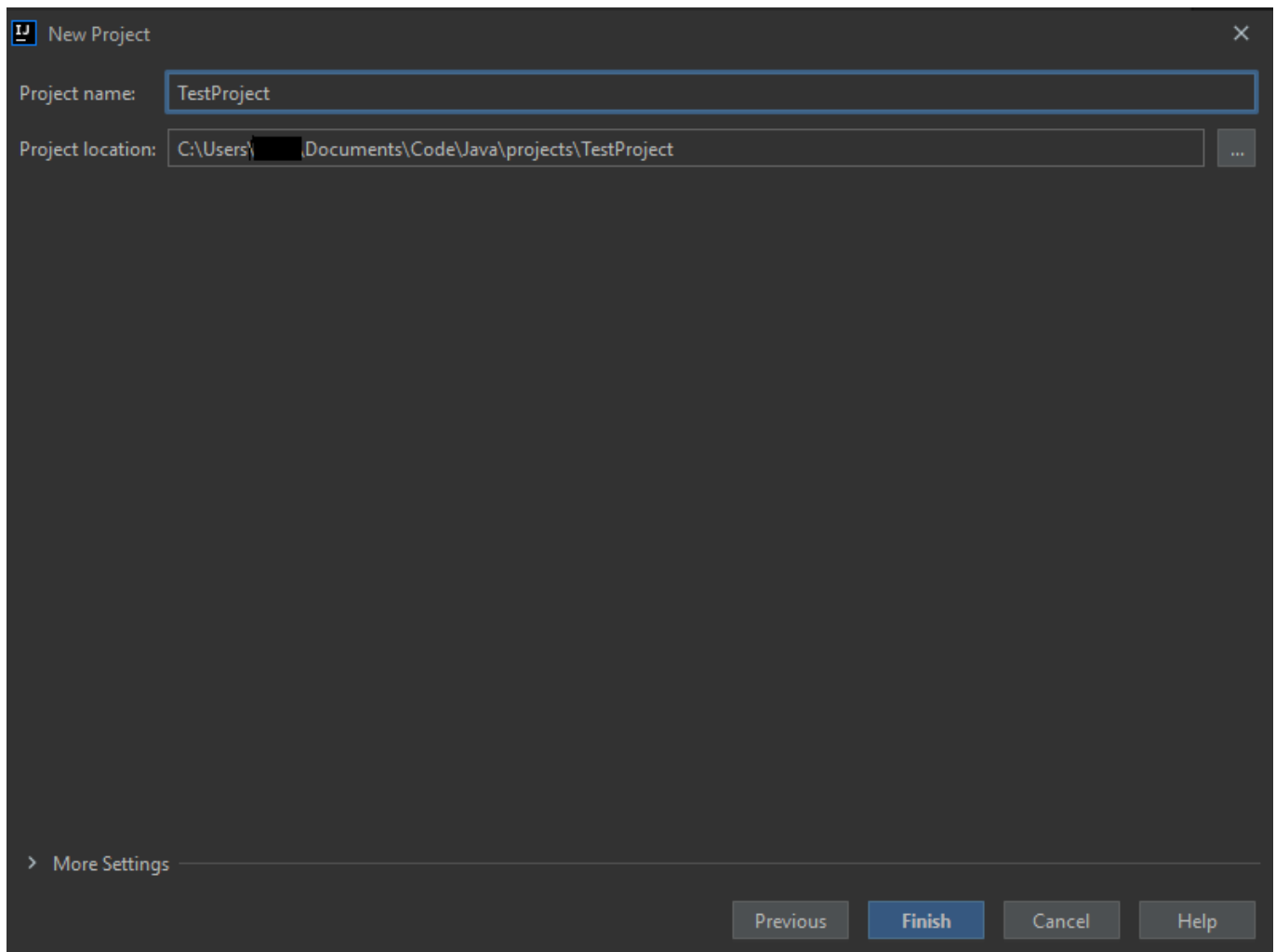
After installing, launch IntelliJ and select **New Project**. If you already had a project open before, you may go to **File > Close Project** (not Exit!) or create a new project directly via **File > New > Project**.



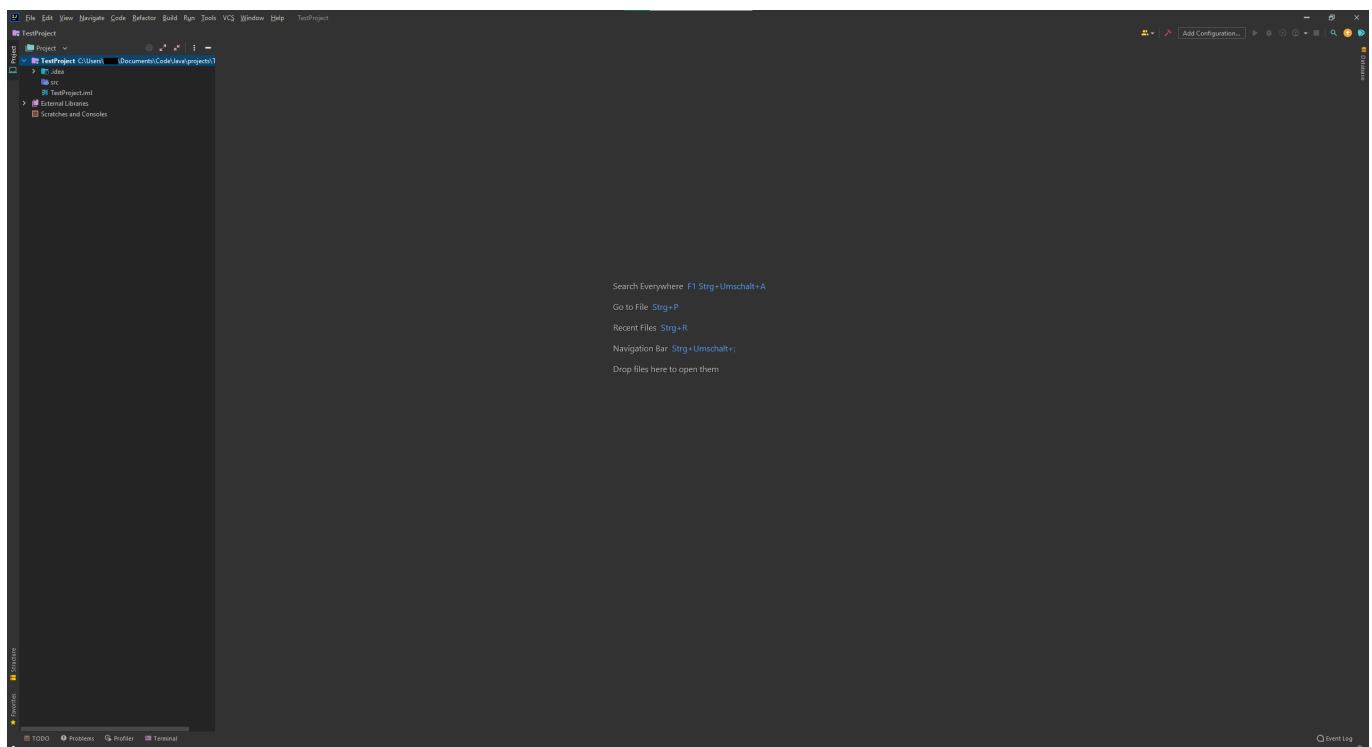
On the left, select **Java**. Open the **Project SDK...** context menu. If you see an option **~\Greenfoot\jdk** listed under defected SDKs, select it. Otherwise, select **Add JDK...**, browse to your Greenfoot installation and select the **jdk** directory.



Select **Next** twice. Now give your project a name and specify its location. Finally, select **Finish**.

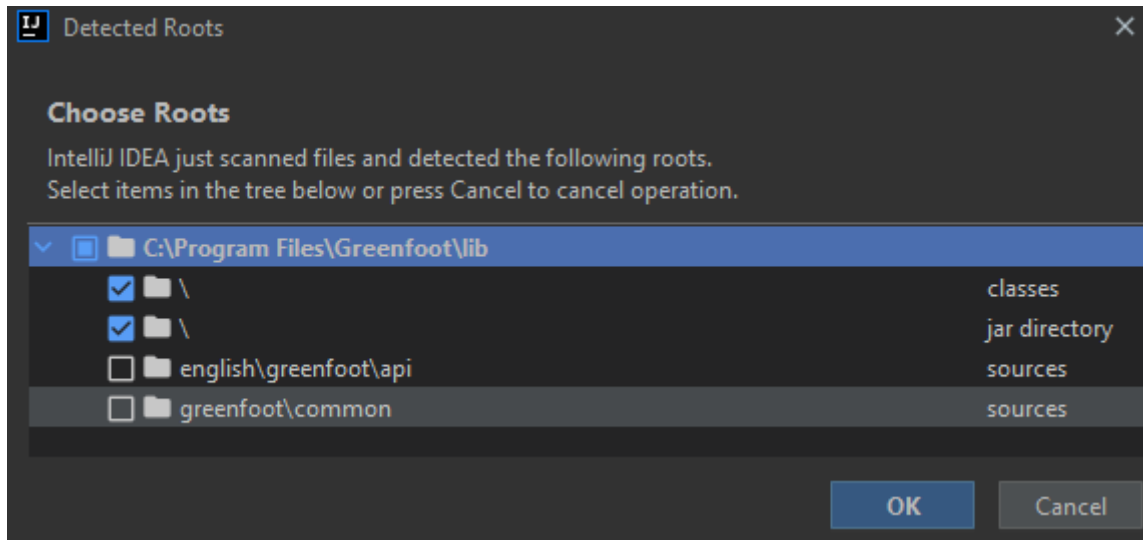


You should now see a window similar to this:



Step 2 - Setup Greenfoot dependencies (only once)

Select **File** > **Project Structure...** > **Global Libraries** > **+** > **Java** and navigate to your Greenfoot installation. Select the **lib** folder and choose the following options:

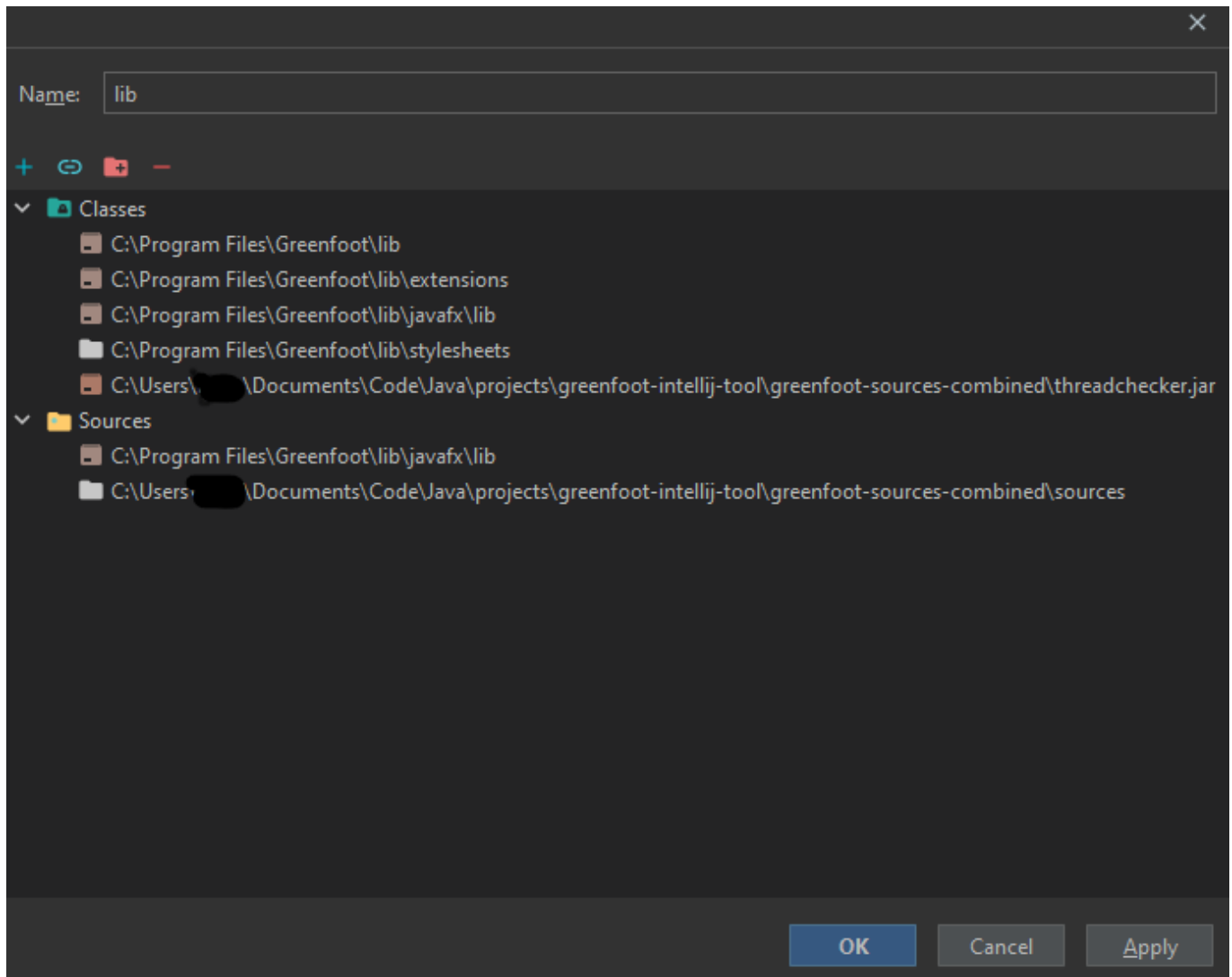


Select **OK** and if you are being asked to add the stuff to modules, also press **OK**.

Use the right **+** button to also add these directories / files:

- `<Greenfoot installation>/lib/extensions`
- `<Greenfoot installation>/lib/javafx/lib`
- `<Greenfoot installation>/lib/stylesheets` (As: Classes)
- `<Combined sources>/sources`
- `<Combined sources>/threadchecker.jar`

The right side of the window should now look something like this:



Give the library a better name like **Greenfoot** and press **OK**.

Step 3 - Add the Greenfoot library to your project

Select **File > Project Structure... > Global Libraries** and right-click onto the library you created. Select **Add to Modules...** and select **OK**. Select **OK** again to save the changes and close the project structure settings.

If you just created the library as in step 2 and got asked whether it should be added to the modules you don't have to do this step and the **Add to Modules...** option will not show up.

Step 4 - Add Greenfoot project files

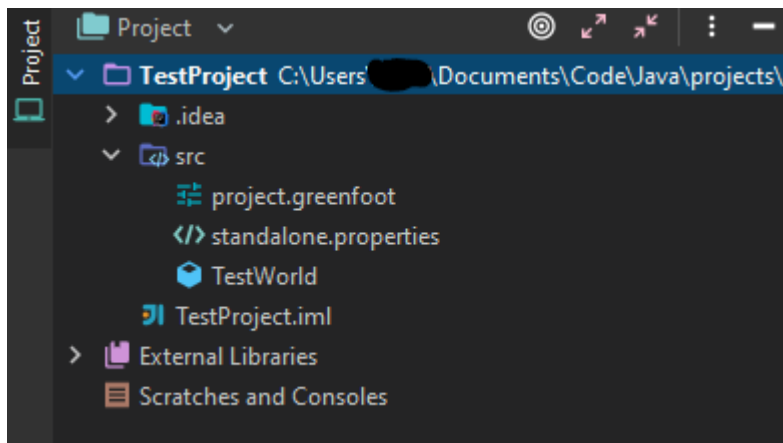
Right-click onto the **src** folder and select **New > File**. Name the file **standalone.properties** and paste the following code into it:

```
main.class=  
controls.pause.button=Pause  
controls.run.button=Start  
run.once=Run
```

```
reset.world=Reset
controls.speed.label=Simulation Speed:
scenario.hideControls=false
controls.speedSlider.tooltip=Adjusts the execution framerate
controls.pause.shortDescription=Pause the simulation.
project.name=Greenfoot scenario made in IntelliJ IDEA
controls.run.longDescription=Run the simulation until stopped.
controls.reset.longDescription=Instantiate a new world.
scenario.viewer.appletInfo=
scenario.lock=false
controls.pause.longDescription=Pause the simulation, leaving it in the current
state.
run.simulation=Run
pause.simulation=Pause
scenario.fullScreen=false
controls.reset.shortDescription=Reset the scenario.
controls.run.shortDescription=Start the simulation.
controls.runonce.longDescription=Call 'act' once for every actor in the current
world.
controls.runonce.shortDescription=Call 'act' once for every actor in the current
world.
user.name=YourNameHere
```

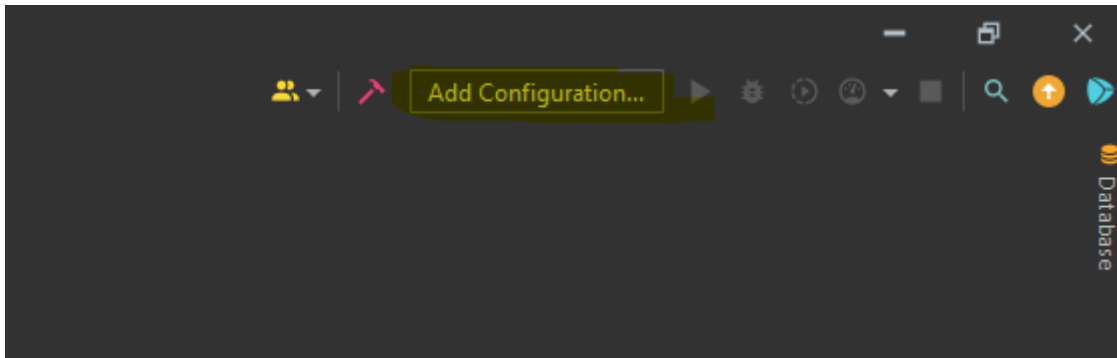
Create another new file named `project.greenfoot` in the `src` folder. You don't need to write anything here, Greenfoot will do this itself. We just create this file so that you can later double-click it from the file explorer to open the project.

Now, your project tree should look like this:



Step 5 - Create launch configuration

In the top right, select `Add Configuration....`

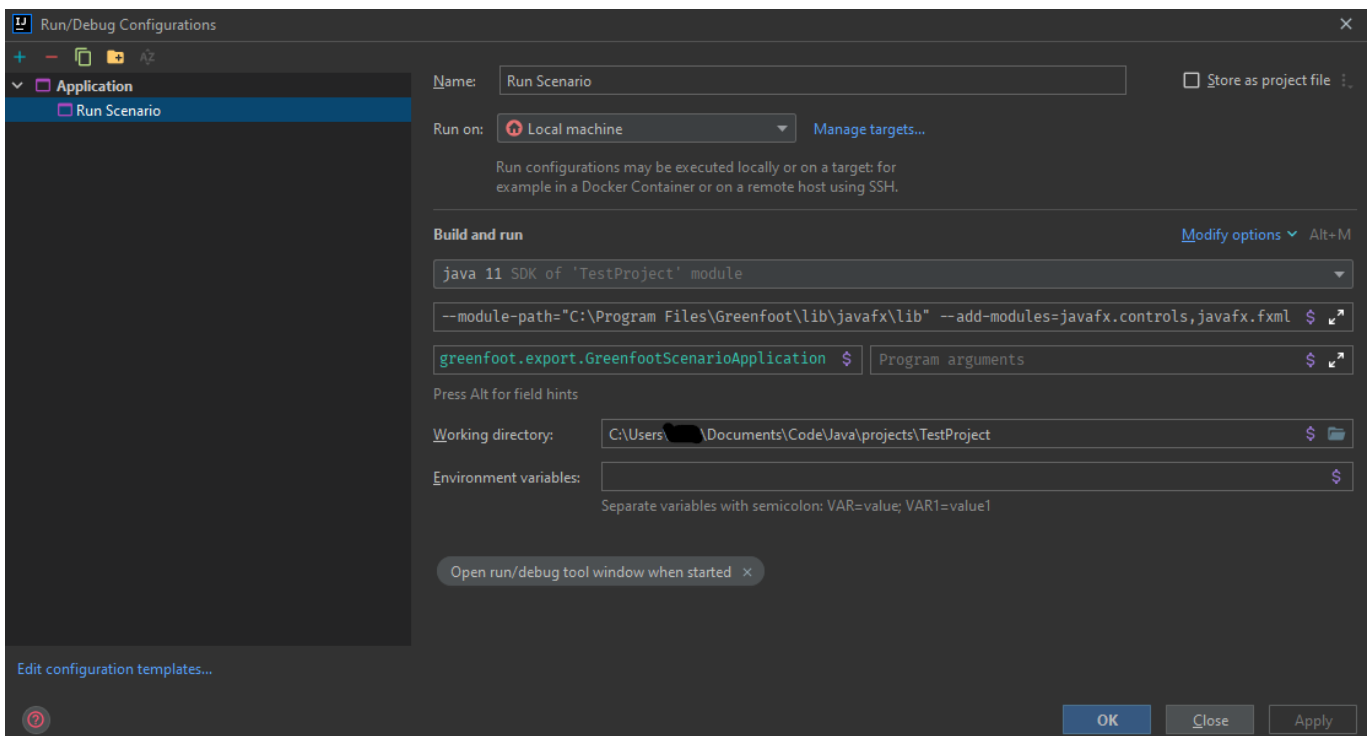


Select **Add new...** > **Application** and give the configuration a proper name like **Run Scenario**. In **Main Class**, enter **greenfoot.export.GreenfootScenarioApplication**.

Select **Modify options** > **Add VM options**. In the text field **VM options** write the following, replacing **<Greenfoot installation>** with the path to your Greenfoot installation folder (the "Greenfoot" folder, not the folder it is in):

```
--module-path="<Greenfoot installation>\lib\javafx\lib"
--add-modules=javafx.controls,javafx.fxml
```

Finally, press **Apply**. Your window should look like this:

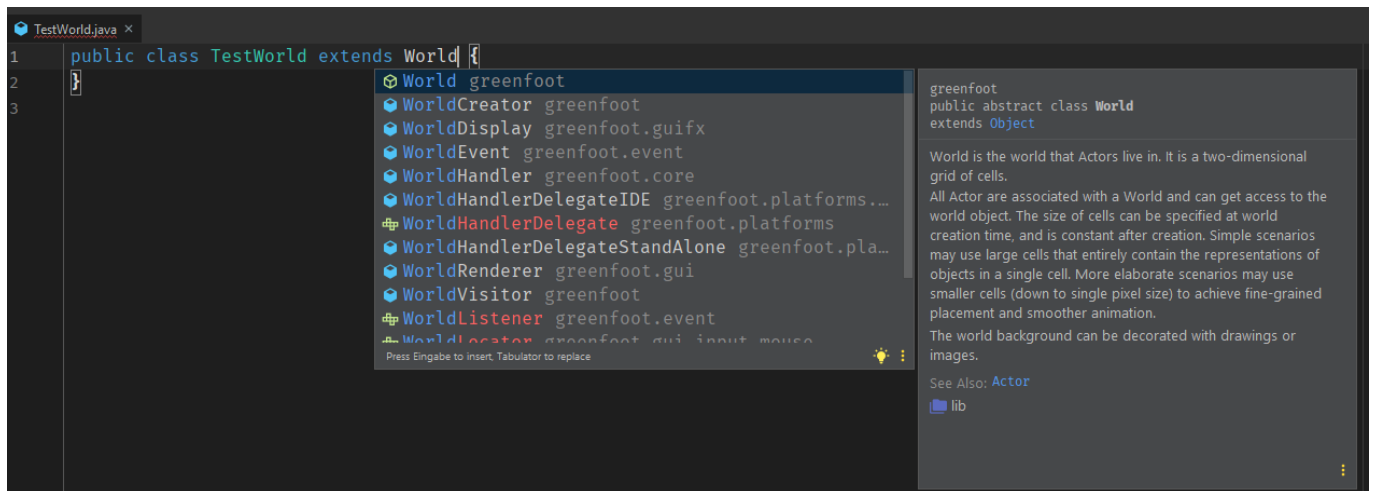


Press **OK**.

Step 6 - Create your first world

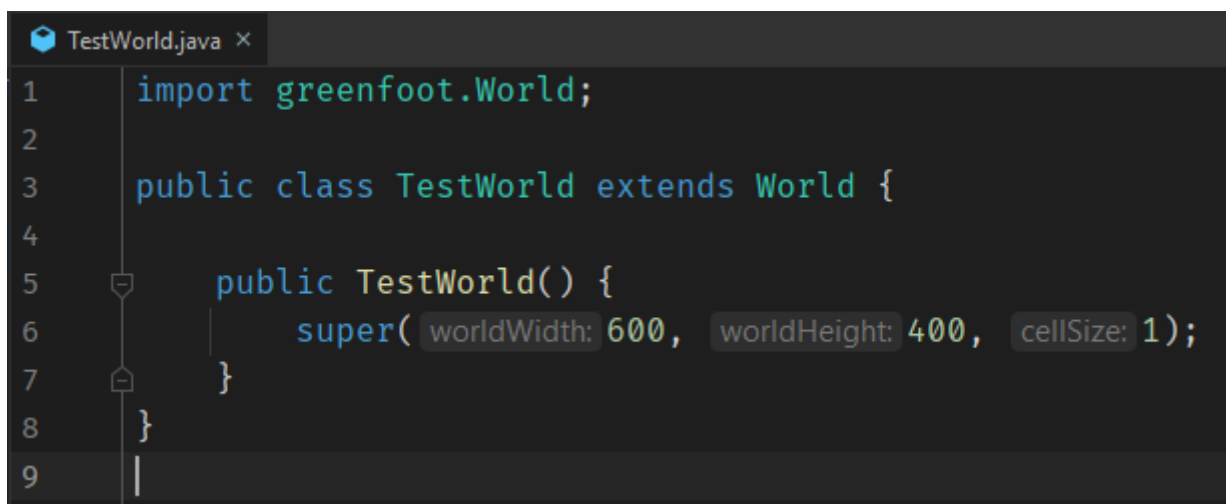
Right-click onto the **src** folder and select **New** > **Java class**. Give your class a valid Java name and press Enter. Place the cursor behind the class name and type **extends World**. After a second or so a completion for

`greenfoot.World` should show up. Press Enter and note how IntelliJ automatically adds the necessary import statement `import greenfoot.World` at the top of the file.



Your code will look a little bit different because I have a custom color theme and font.

You next need to add a constructor to your world, like usually generated by Greenfoot. It looks something like this:



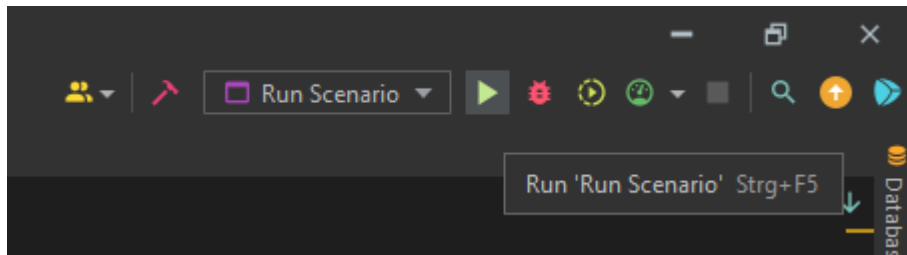
Note that the gray blocks with `worldWidth`, `worldHeight` and `cellSize` are inserted by IntelliJ. Do **not** write them yourself! Also, this info may not show up for you, I've activated them in the settings.

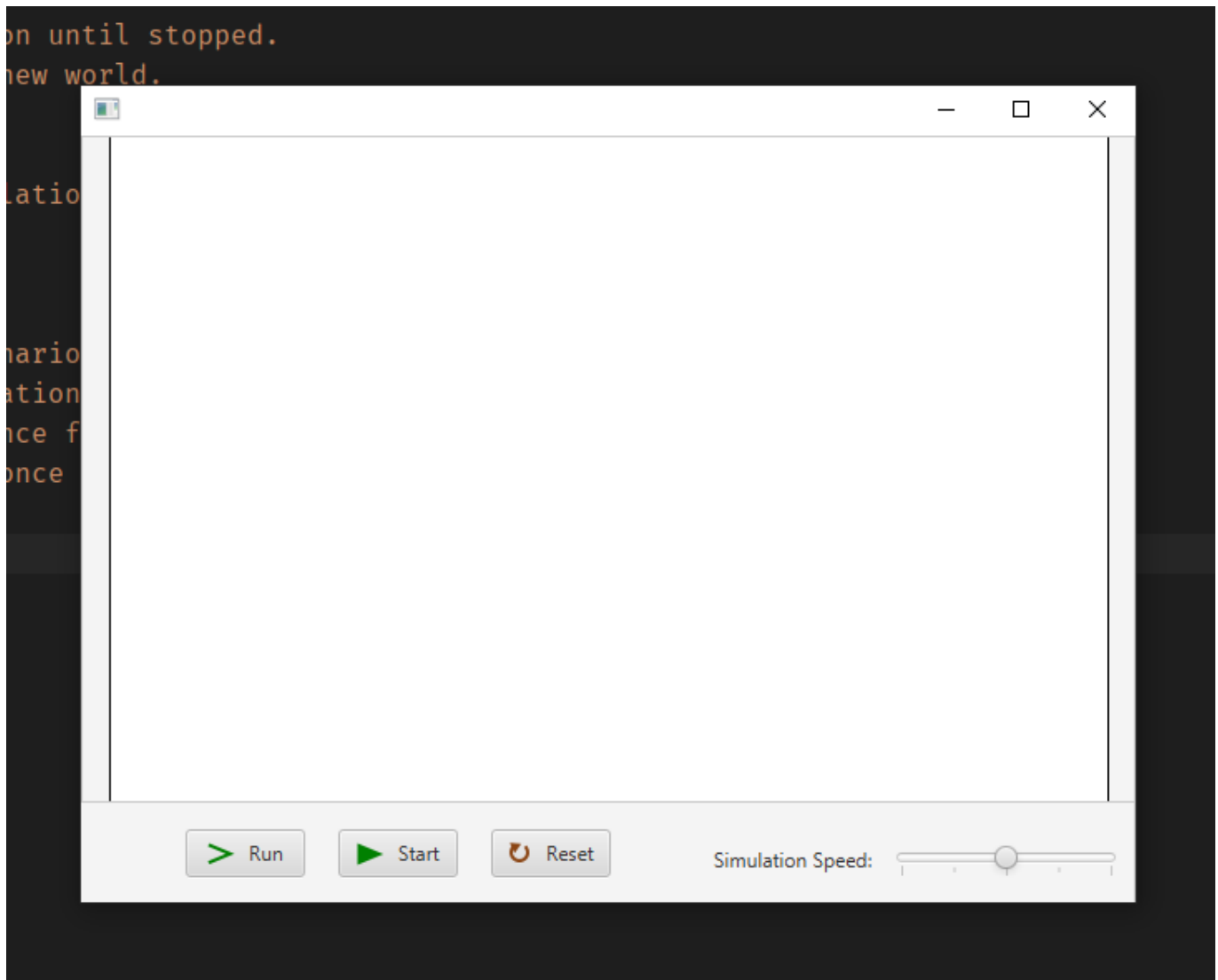
Usually you would now exit the Greenfoot editor, right-click onto your world and select `new <YourWorld>()`. When using IntelliJ, you specify the start world in the `standalone.properties` file. In the first line, append the name of your World.

```
TestWorld.java x standalone.properties x project.greenfoot x
1  main.class=TestWorld
2  controls.pause.button=Pause
3  controls.run.button=Start
4  run.once=Run
5  reset.world=Reset
6  controls.speed.label=Simulation Speed:
7  scenario.hideControls=false
8  controls.speedSlider.tooltip=Adjusts the execution framerate
9  controls.pause.shortDescription=Pause the simulation.
10 project.name=Greenfoot scenario made in IntelliJ IDEA
11 controls.run.longDescription=Run the simulation until stopped.
12 controls.reset.longDescription=Instantiate a new world.
13 scenario.viewer.appletInfo=
14 scenario.lock=false
15 controls.pause.longDescription=Pause the simulation, leaving it in the current state.
16 run.simulation=Run
17 pause.simulation=Pause
18 scenario.fullScreen=false
19 controls.reset.shortDescription=Reset the scenario.
20 controls.run.shortDescription=Start the simulation.
21 controls.runonce.longDescription=Call 'act' once for every actor in the current world.
22 controls.runonce.shortDescription=Call 'act' once for every actor in the current world.
23 user.name=RcCookie
24
```

Step 7 - Run your scenario

In the top-left corner, press the Run button. A window should open with the usual Greenfoot controls showing your world.





Now close the window and start coding!

(Step 8 - Open the project in the Greenfoot application)

Normally you should not need the Greenfoot application anymore while coding. However, there are still some cases in which it is needed, in specific when you want to upload your scenario to the Greenfoot website.

Opening your project in the Greenfoot application can be done no problem. Just open the `src` folder of your project in the file explorer, i.e. by right-clicking onto the folder in IntelliJ and selecting `Open in > Explorer`. Now just double-click the `project.greenfoot` file. Greenfoot will show a message that the file is from an older version, just click ok. This is because we did not write anything to the file ourselves. When the scenario has loaded for the first time into the application, you will have to right-click onto your main world and select `new <YourWorld>()`. Now your scenario should be running and can be exported the usual way.