NetBeans Tutorial

What’s NetBeans?

- Like Eclipse:
  - It is a free software / **open source** platform-independent software tool for delivering what the project calls "rich-client applications"
  - It is an **Integrated Development Environment** (IDE), that allows to manage the whole development process of Java applications, by providing many features for programming (editor, debugger, etc.)
  - It supports other languages by means of plug-ins (C/C++)
  - Multi-platform (Linux, Windows, Mac OS)

What’s NetBeans?

- NetBeans was originally developed at the Faculty of Mathematics and Physics at Charles University in Prague .
- Then it was bought by Sun Micosystems, then Oracle.
- NetBeans is now managed by the Apache Software Foundation, a decentralized open source community of developers. The software they produce is distributed under the terms of the Apache License and is free and open-source software (FOSS)

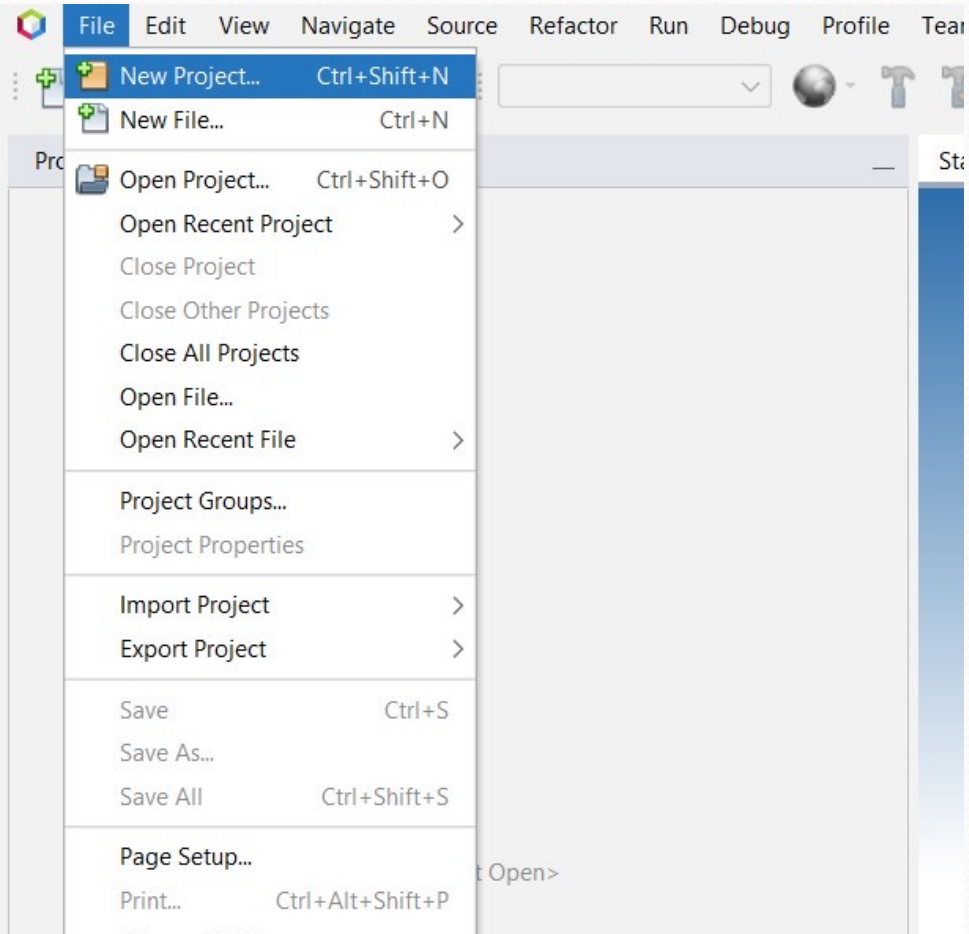
Getting NetBeans

- On your laptop
  - You will need to install a Java Virtual Machine (JVM)
  - Download the latest version at:
    - https://netbeans.apache.org/download/index.html (Select your OS and language, then download the Java SE version)

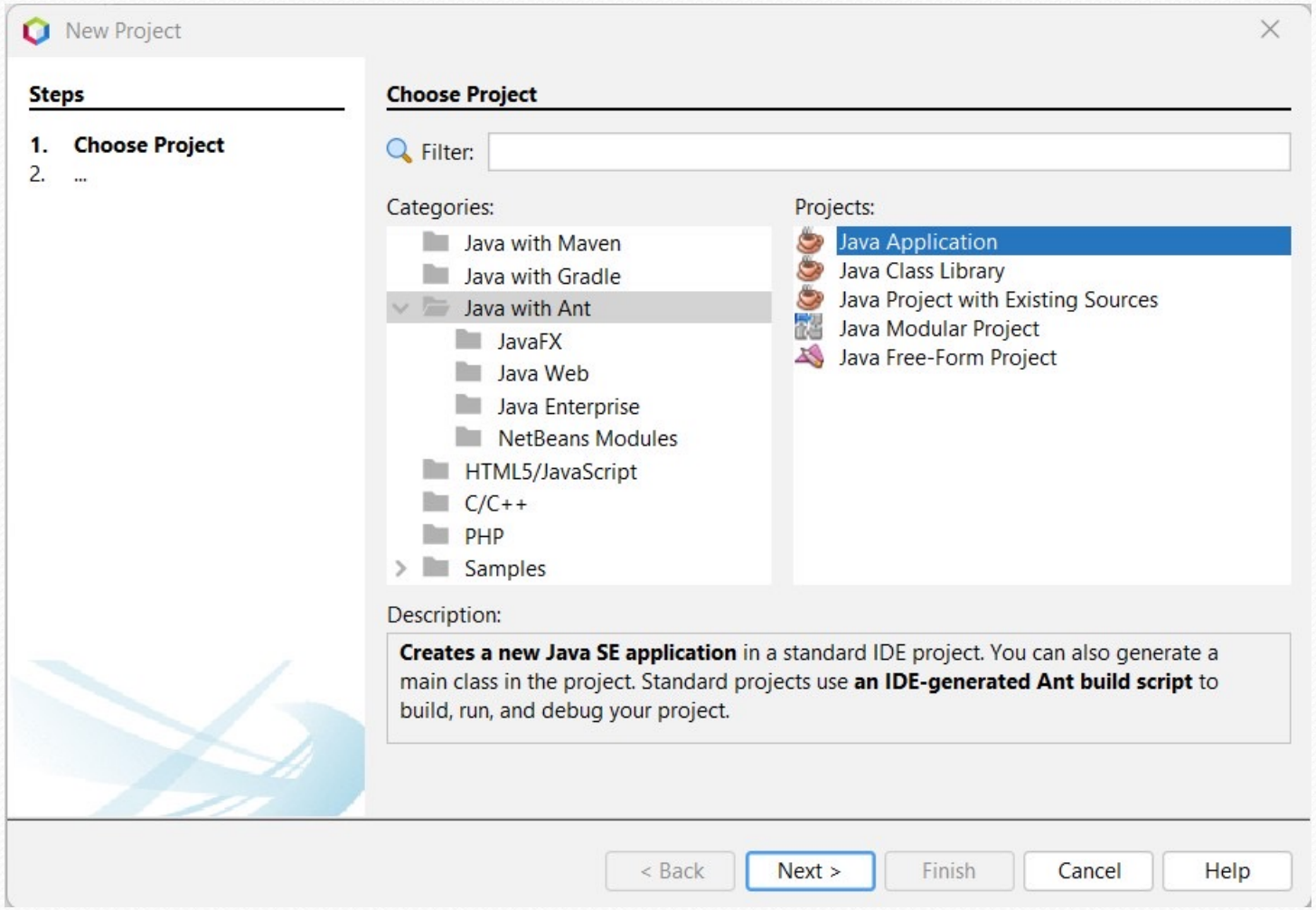
Let’s start with basic stuff

Step1: Open NetBeans from start on your system.

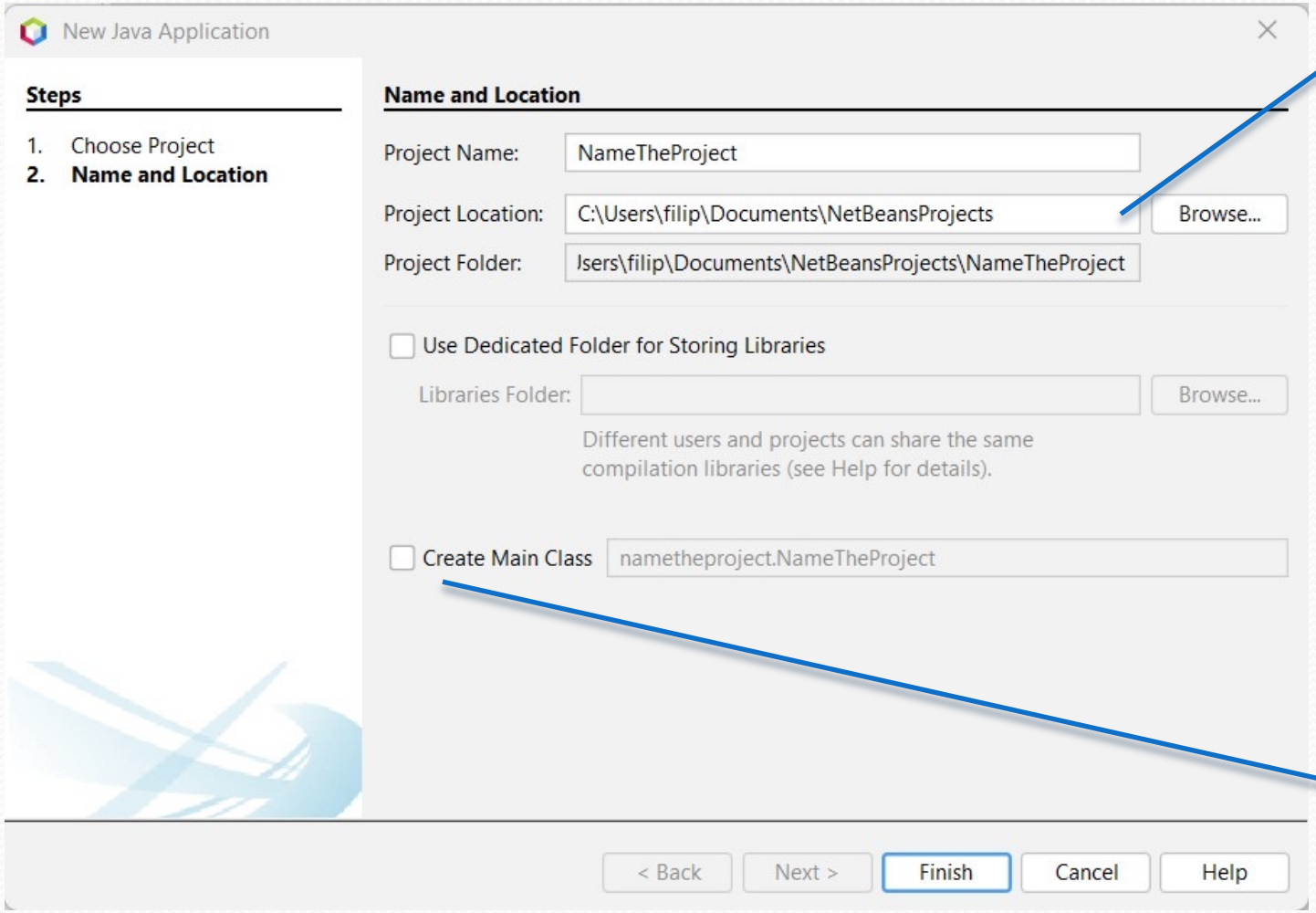
Step2: Create a new project (File → New Project…).



Step3: Select *Java with Ant-> Java Application* and click next.



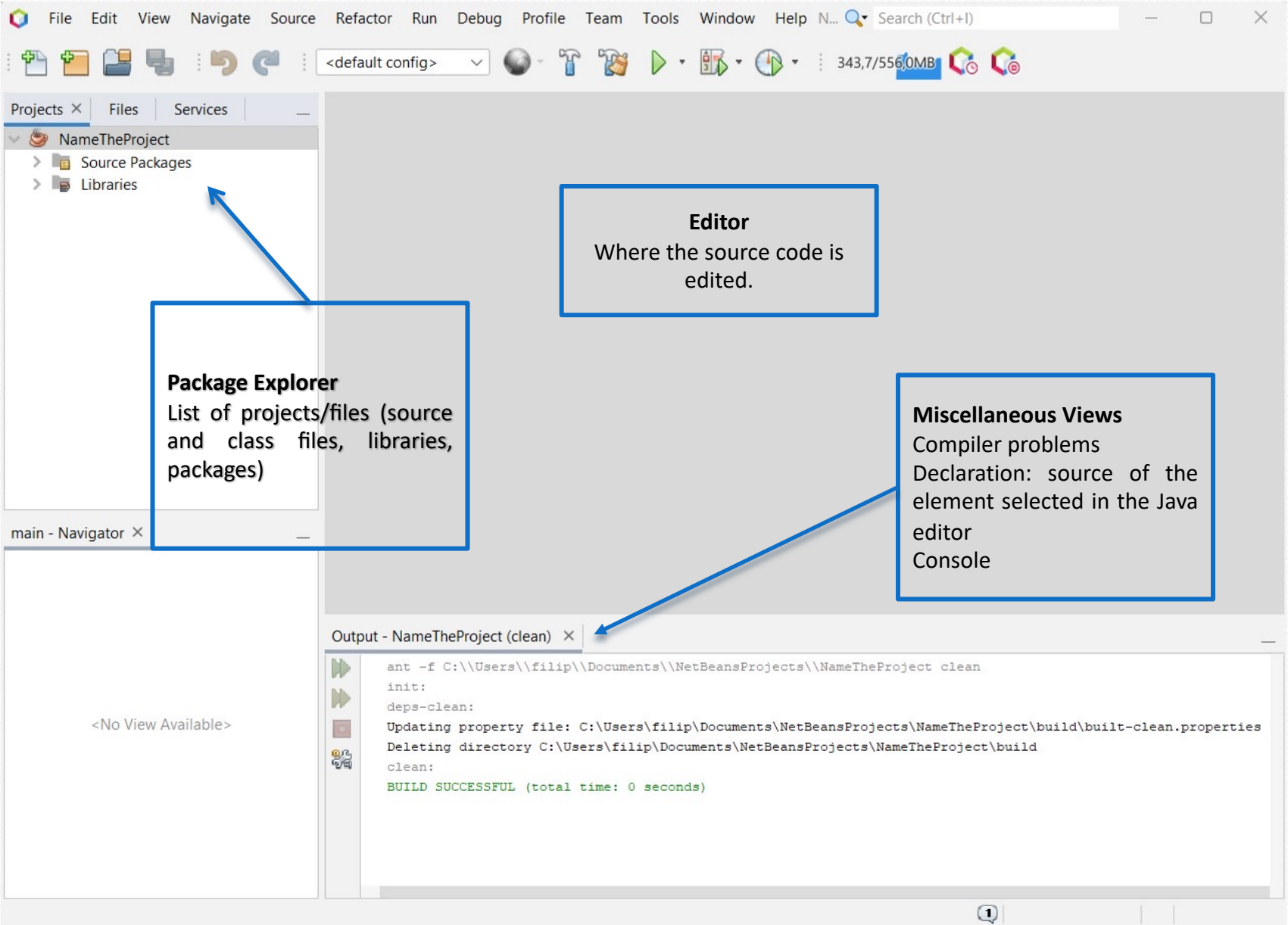
Step4: Name the project and click finish.



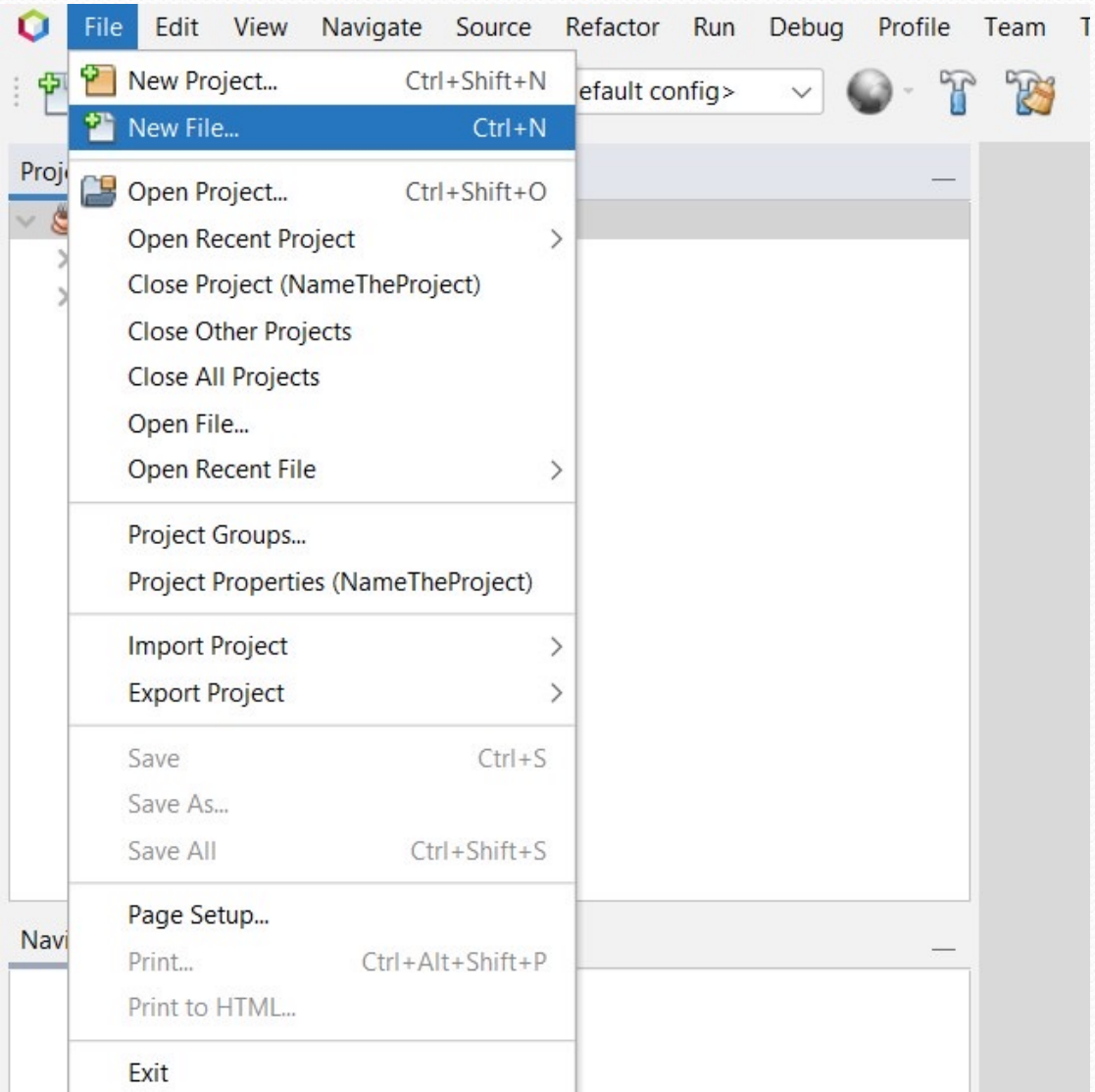
Creates the directory with the specified «project name» in the workspace path.

Deselect the option *Create Main Class*, in this way we will create an empty project.

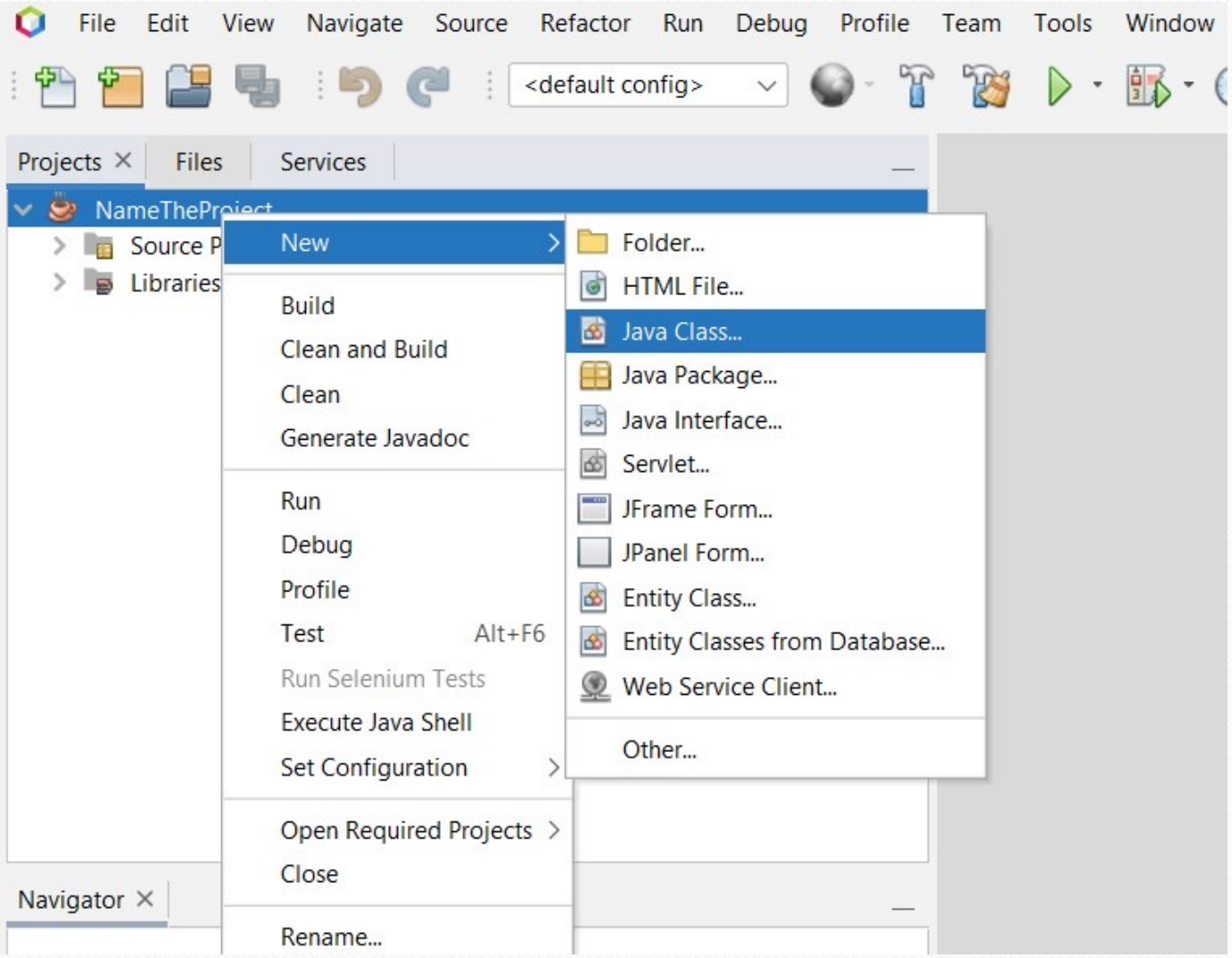
Step5: IDE views



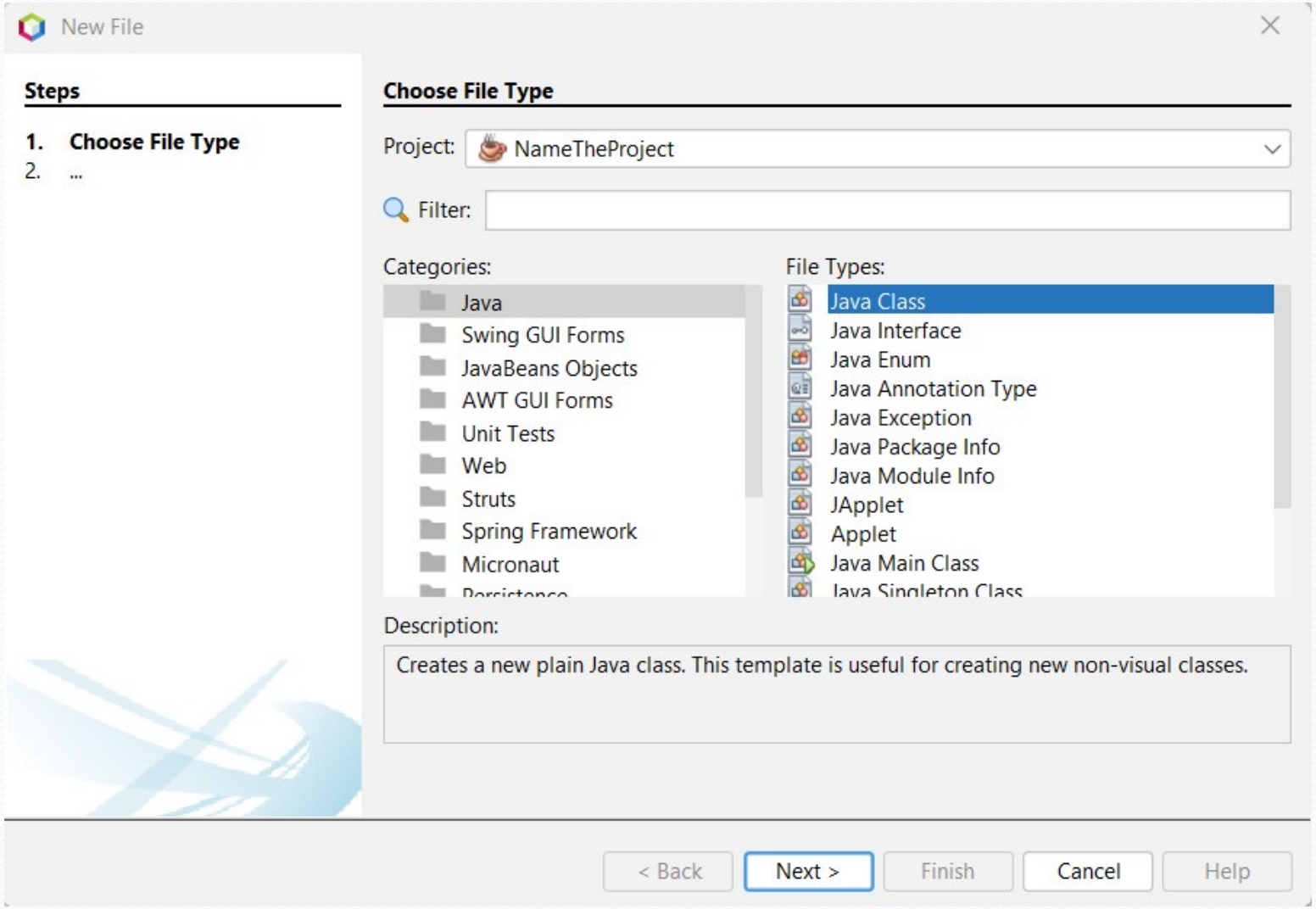
Step6: Now we create the java file by selecting the “File” menu, then “New File…”.



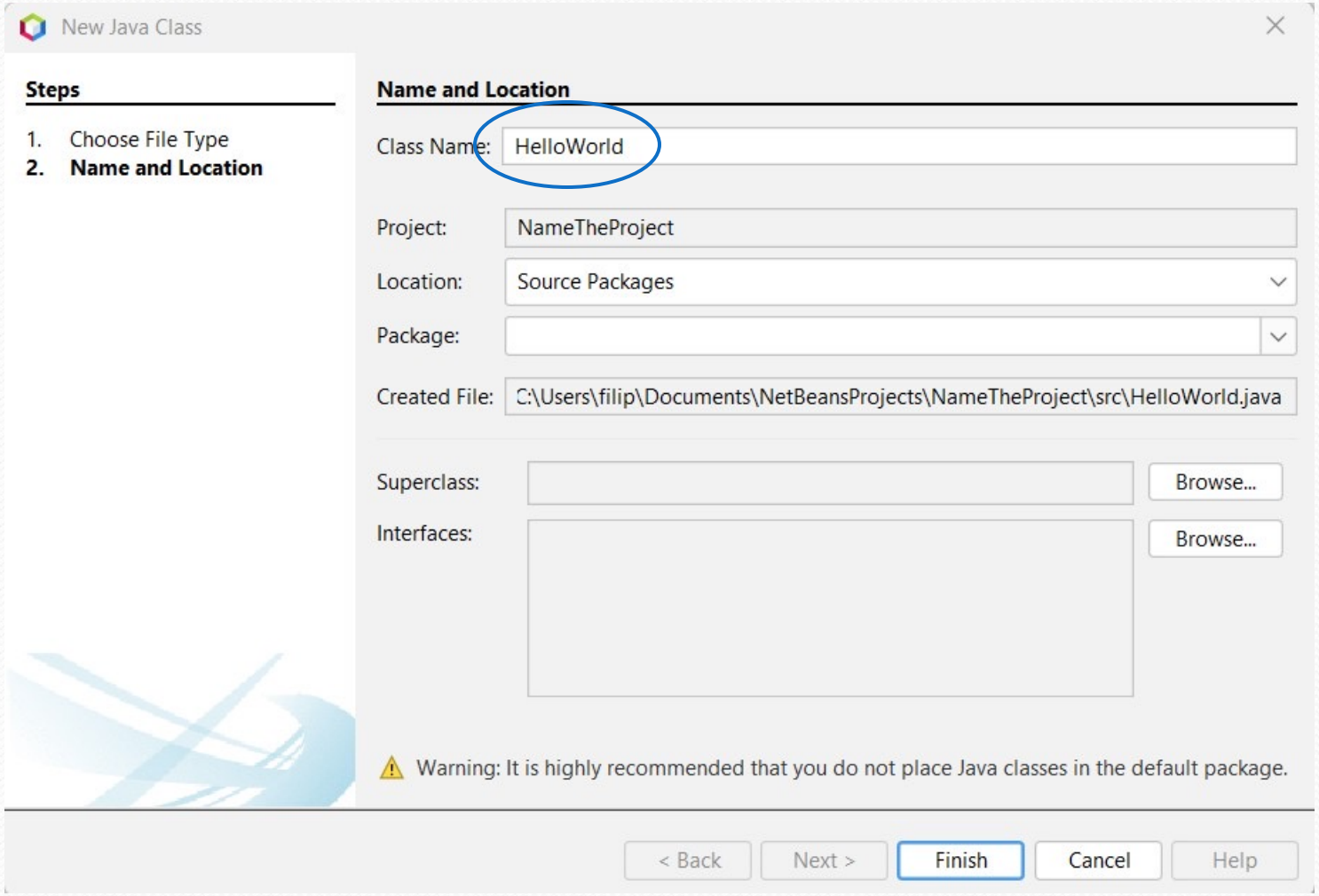
Step6 bis: Or else, we can right-click on the project in the project manager and select “New”, then “Java Class…”.



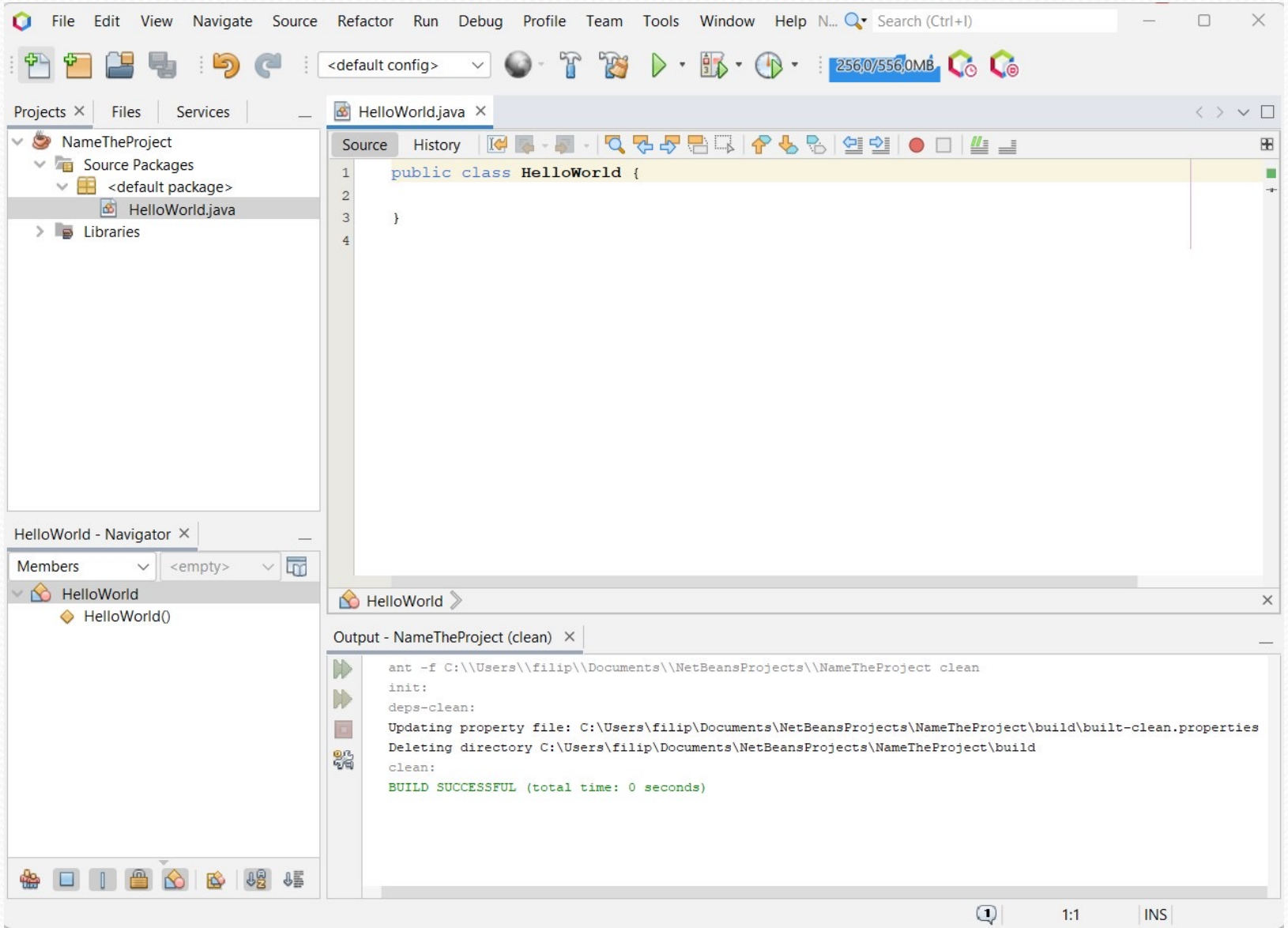
Step7: Now select the project in which you want to write and the type of file you want to add. In the example Java Class is selected.



Step8: Now chose the name of the class you want create and click “Finish”.

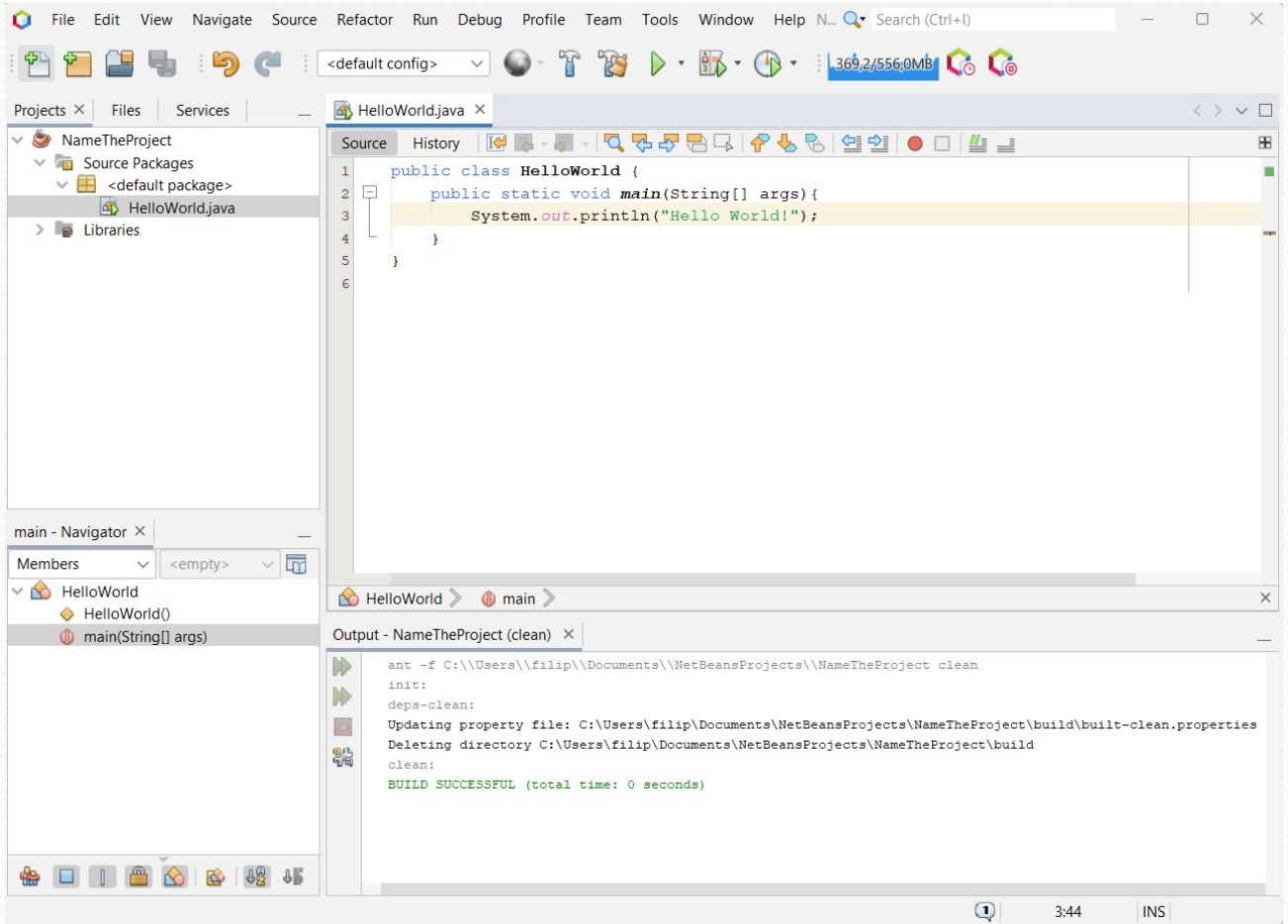


Step9: Now you have the editor space, start coding.

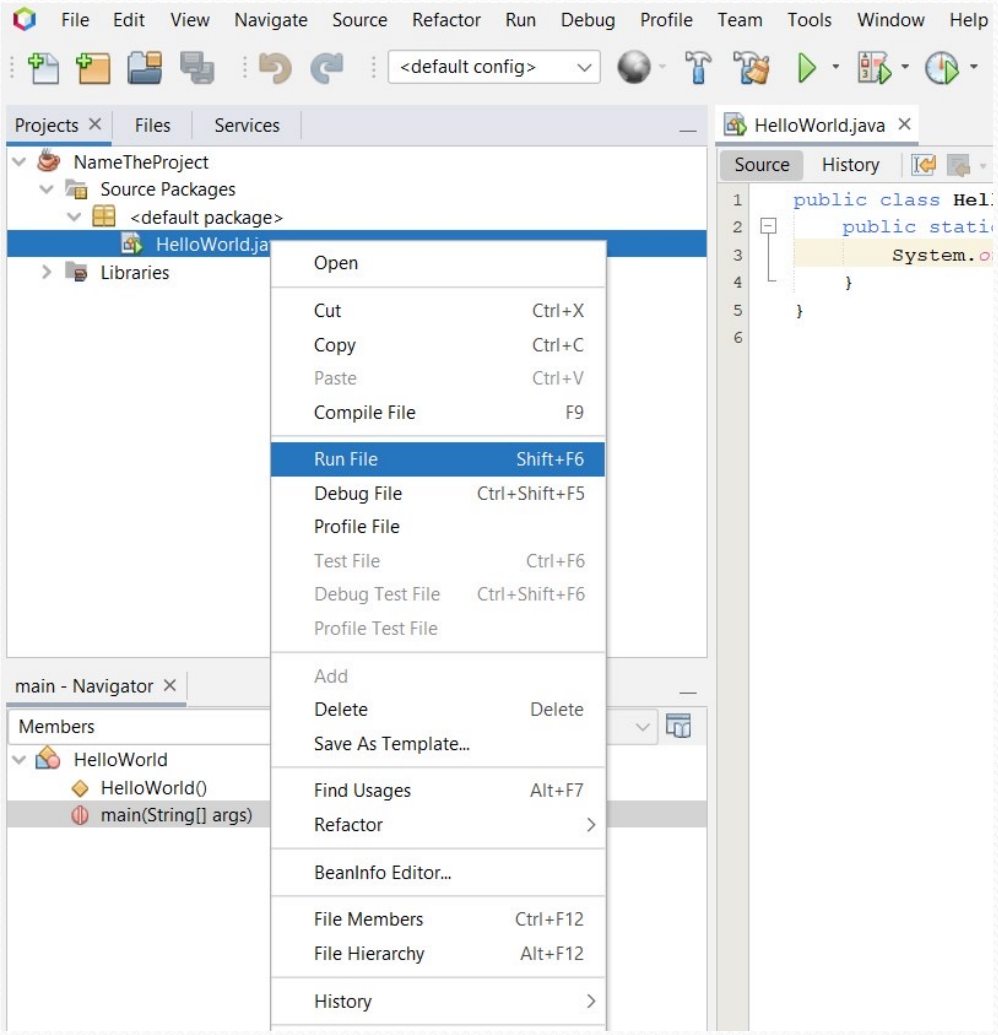


Writing the code

Step10: In NetBeans when ever you save the file, it will compile the code by default.



Step11: Running the java class. Right click on the class file and choose “Run File”.



Step12: Here you can find the output (Console).

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

NameTheProject
Source Packages
<default package>
HelloWorld.java
Libraries

HelloWorld.java

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

main - Navigator

Members

HelloWorld
HelloWorld()
main(String[] args)

Output - NameTheProject (run)

```
run:
Hello World!
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output

Features of NetBeans

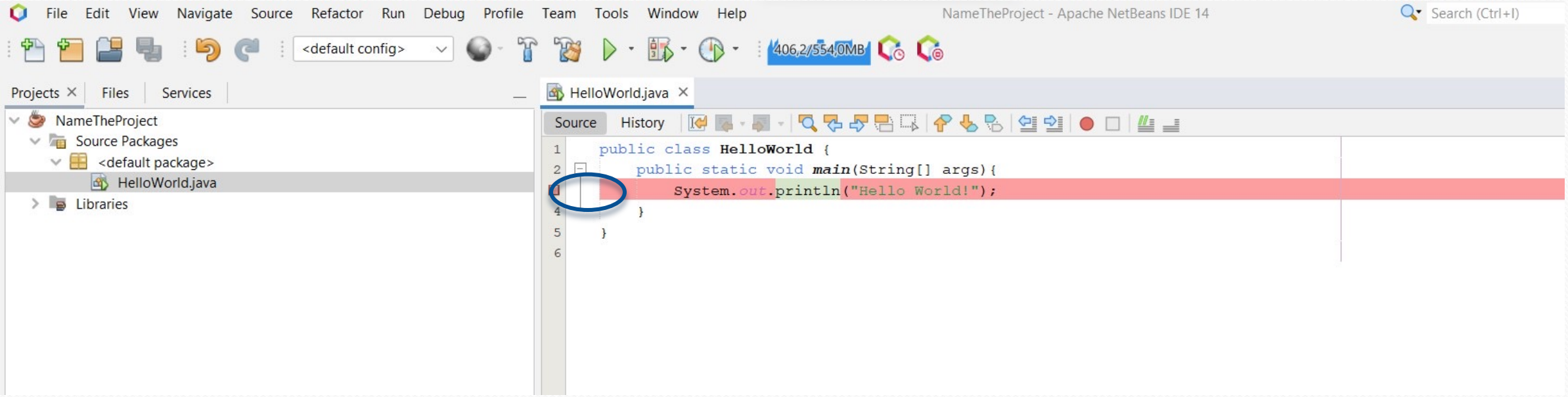
- NetBeans has the basic features required for editing, running, and debugging Java code. In addition to basic programming features, it support for **more advanced Java development tools** such as Ant, Maven, gradle, git, JUnit, and refactoring.

Running code

- NetBeans uses an incremental compiler, so it isn't necessary to explicitly compile your Java files; **the compiled class files are saved automatically when you save your Java files**. However, it is possible to force recompiling by selecting Clean and Build from the menu opened by right-clicking on the project name.
- To run a program, the easiest way is to **select the file containing a main()** method in the Package Explorer and then select **Run File** from the main NetBeans menu or clicking on the big green triangle in the toolbar.

Debugging

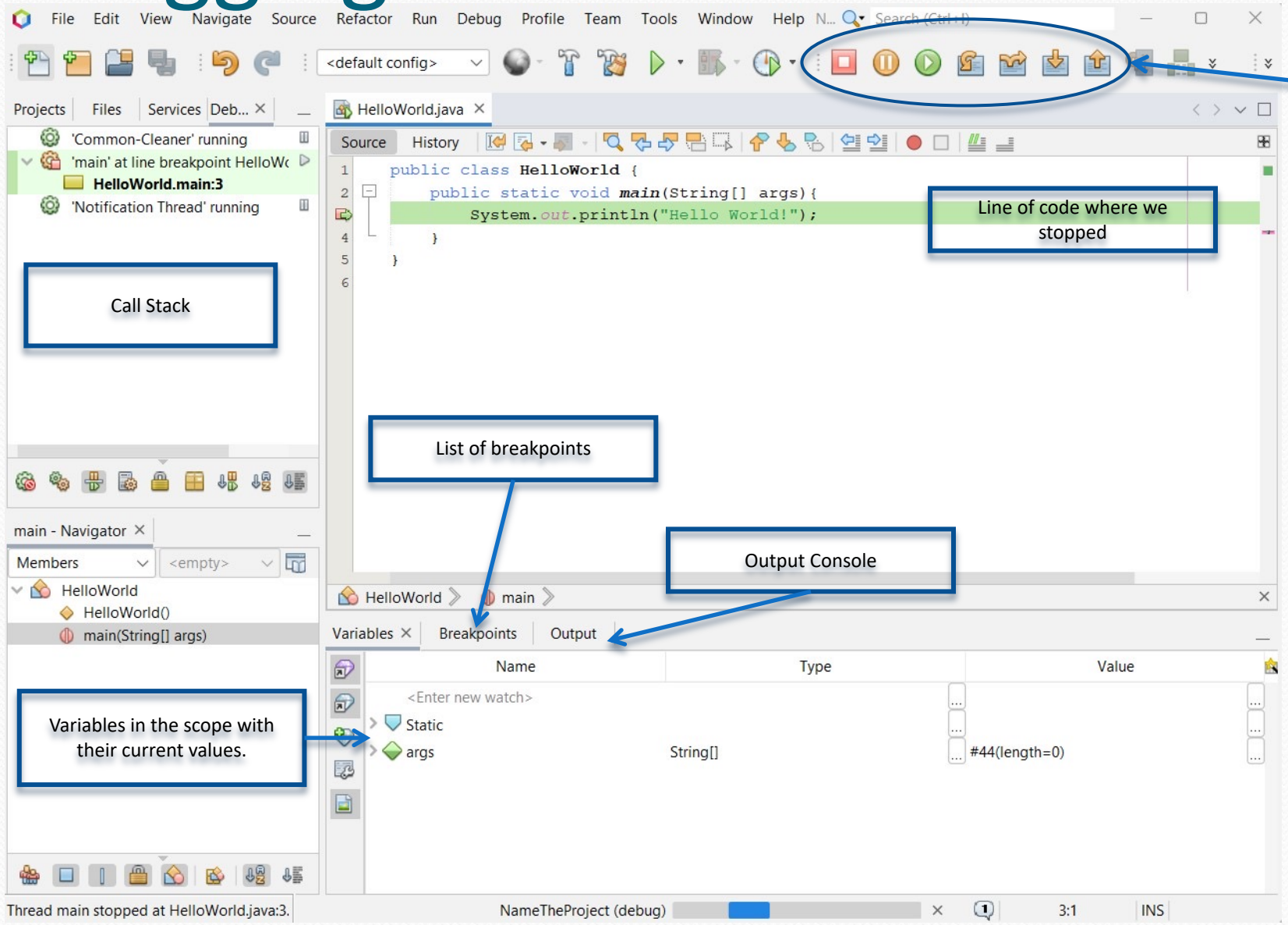
- First, set a breakpoint in the main() method by clicking in the left margin next to the call. If this code were a little less trivial, it would also be possible to set a conditional breakpoint -- one that stops when a particular expression is true, or one that stops after a specific number of hits -- by right-clicking the breakpoint and selecting **Method Breakpoint > properties** from the context menu.



Debugging

- To start debugging, select **Debug > Debug Project** or **Debug > Debug File** from the main menu. Some new toolbars and windows will be opened to show breakpoints and allows step-by-step execution.

Debugging



Buttons to step thorugh the code. Shortcuts available from Run menu

Step **into**

- Step **into** will cause the debugger to descend into any method calls on the current line. If there are multiple method calls, they'll be visited in order of execution; if there are no method calls, this is same as step over. This is broadly equivalent to following every individual line of execution as would be seen by the interpreter.

Step **over**

- Step **over** proceeds to the next line in your current scope (i.e. it goes to the next line), without descending into any method calls on the way. This is generally used for following the logic through a particular method without worrying about the details of its collaborators, and can be useful for finding at what point in a method the expected conditions are violated.

Step **out**

- Step **out** proceeds until the next "return" or equivalent - i.e. until control has returned to the preceding stack frame. This is generally used when you've seen all you need to at *this* point/method, and want to bubble up the stack

The famous Dos and Don’ts:

- Have the project folder where you can easily access it (normally they are saved in the NetBeans Projects folder in Documents).
- **Never start writing the code without making a project**. You need to create a project folder every time you start a new assignment
- **Main classname and the file name** should always match.