NetBeans Tutorial

What’s NetBeans?

- Like Eclipse:
  - It is a free software / **open source** platform-independent software framework 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.
- NetBeans is now managed by Oracle, specialized in developing and marketing database software and technology, cloud engineered systems and enterprise software products.

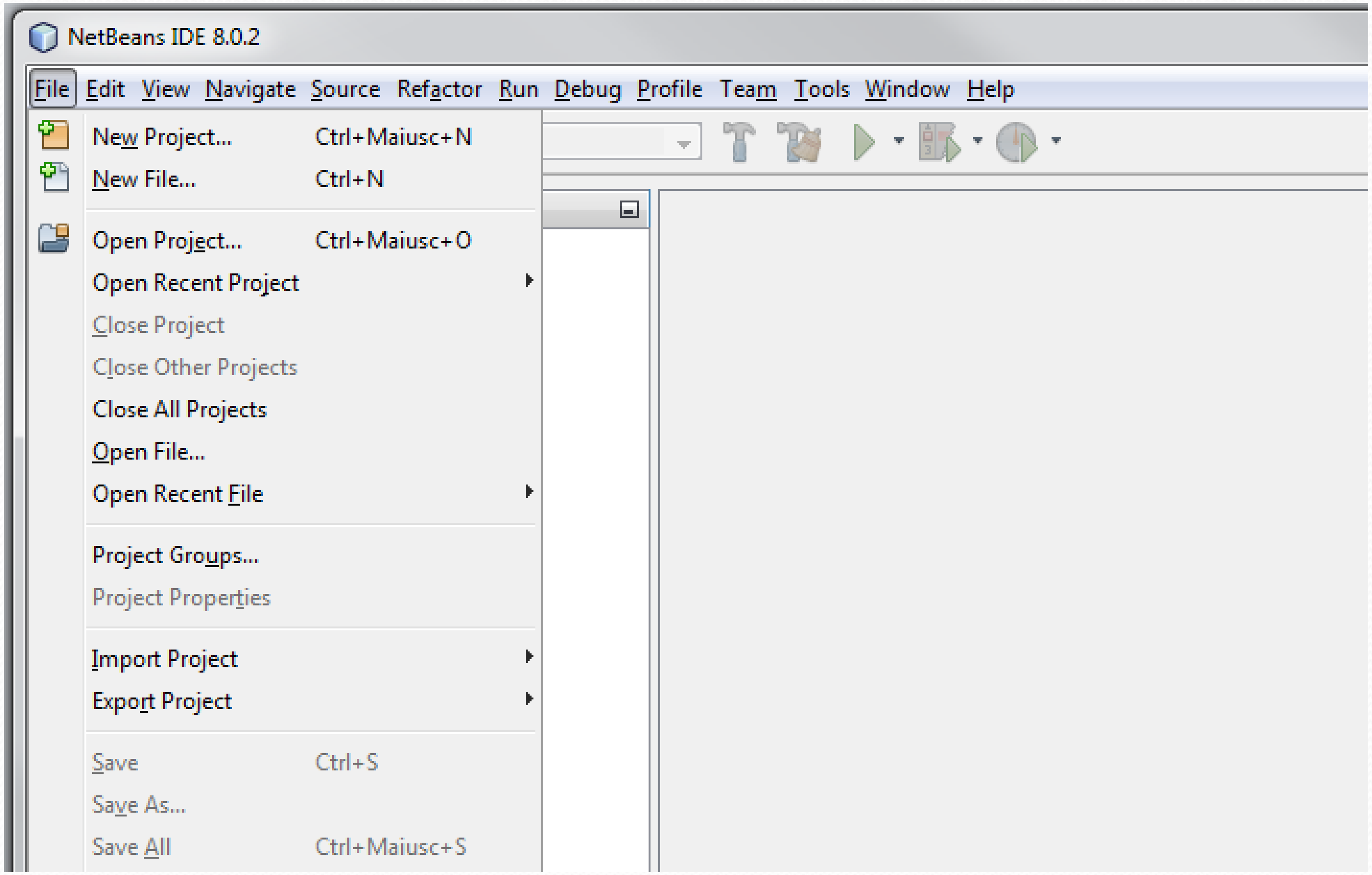
Getting NetBeans

- On your laptop
  - You will need to install a Java Virtual Machine (JVM)
    - http://www.oracle.com/technetwork/java/javase/downloads/index.html (JDK)
  - Download the latest version at:
    - https://netbeans.org/downloads/ (Select your OS and language, then download the Java SE version)
    - Or, select OS independent ZIP, decompress the downloaded archive and run the executable located at **`netbeans/bin/netbeans`**
  - Installation steps (for the last version) at https://netbeans.org/community/releases/80/install.html

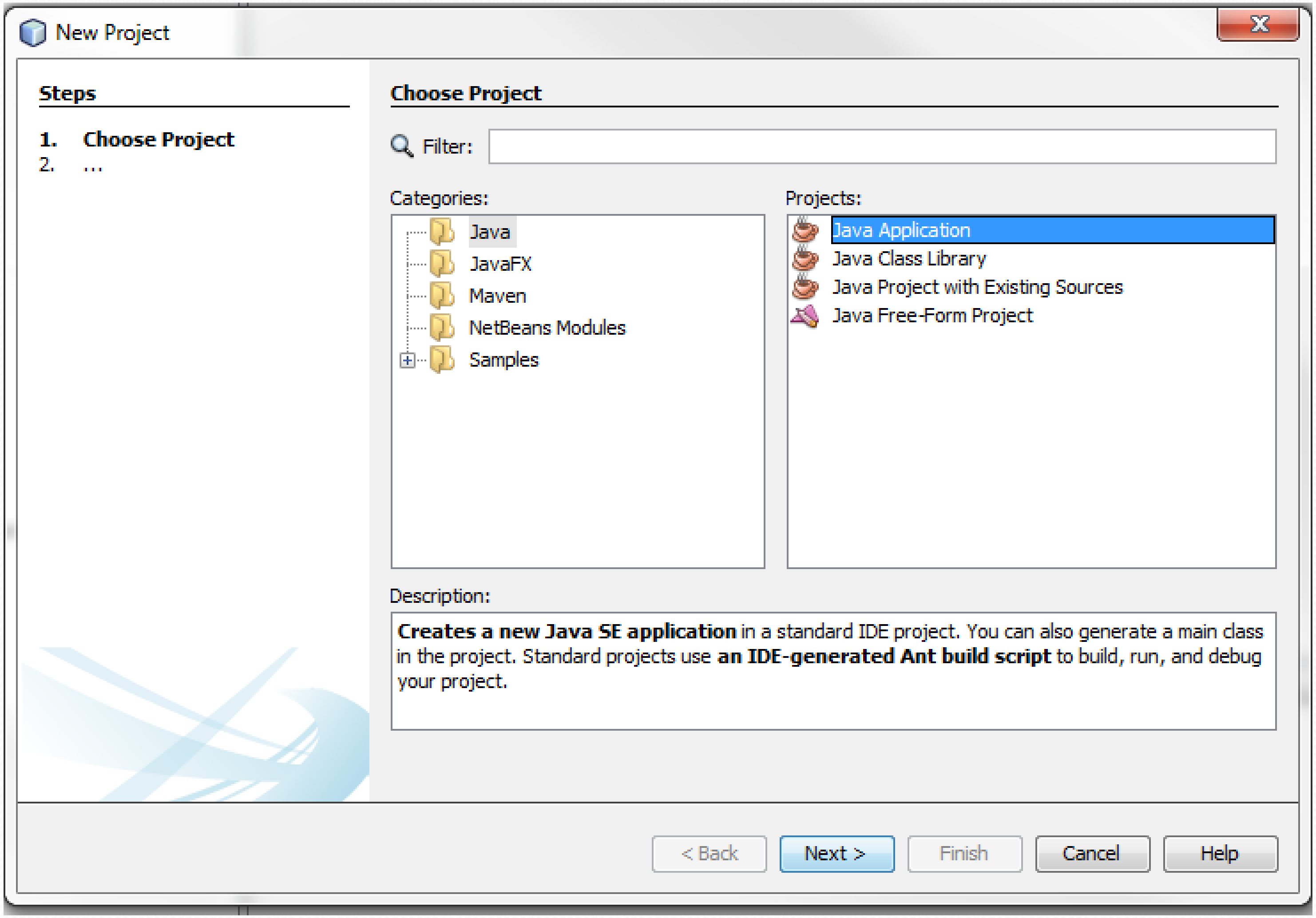
Let’s start with basic stuff

Step1: Open NetBeans from start on your system.

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

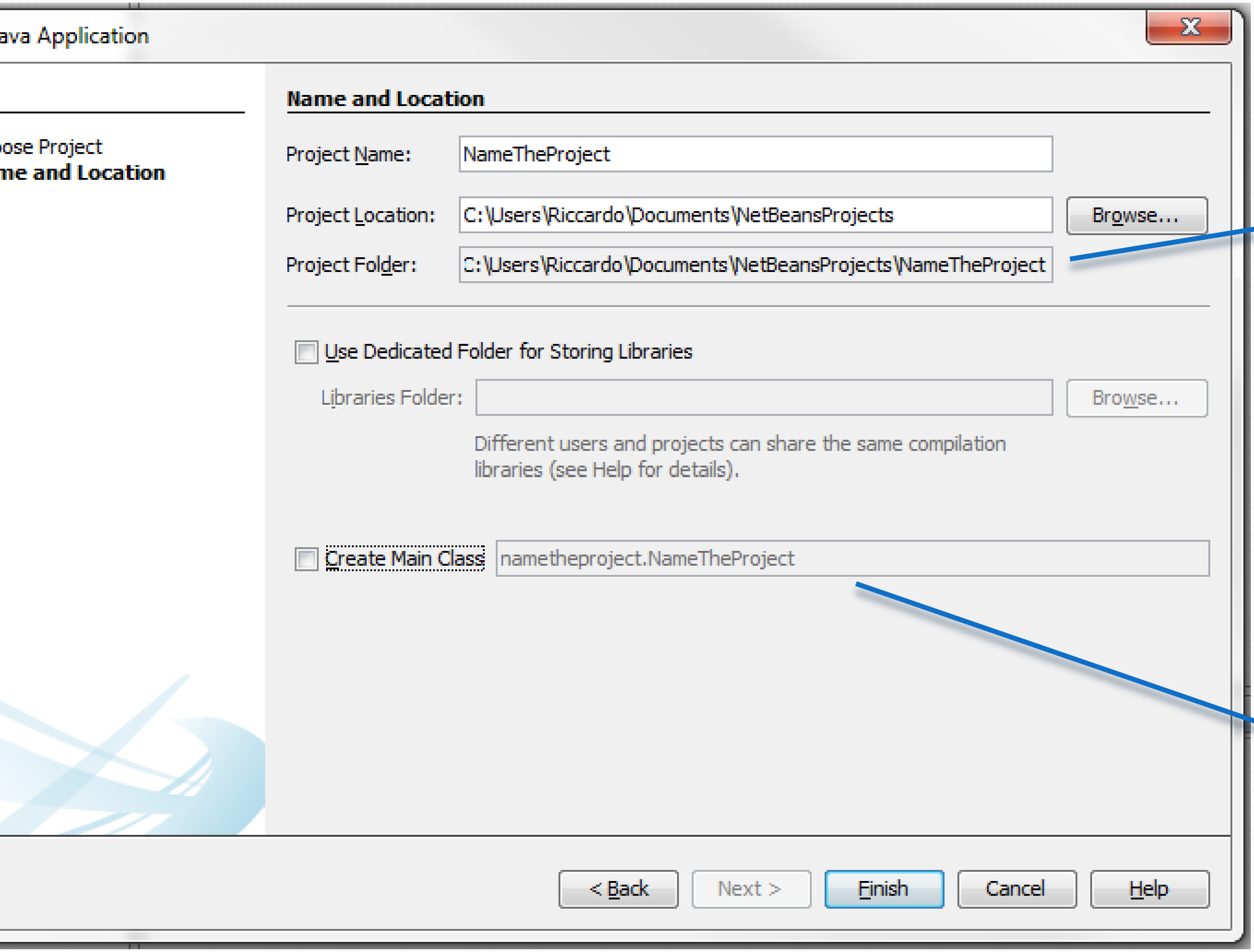


Step3: Select *Java Application* and click next.



Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Step4: Name the project and click finish.



ava Application

ose Project
me and Location

Name and Location

Project Name: NameTheProject

Project Location: C:\Users\Riccardo\Documents\NetBeansProjects

Project Folder: :\Users\Riccardo\Documents\NetBeansProjects\NameTheProject

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

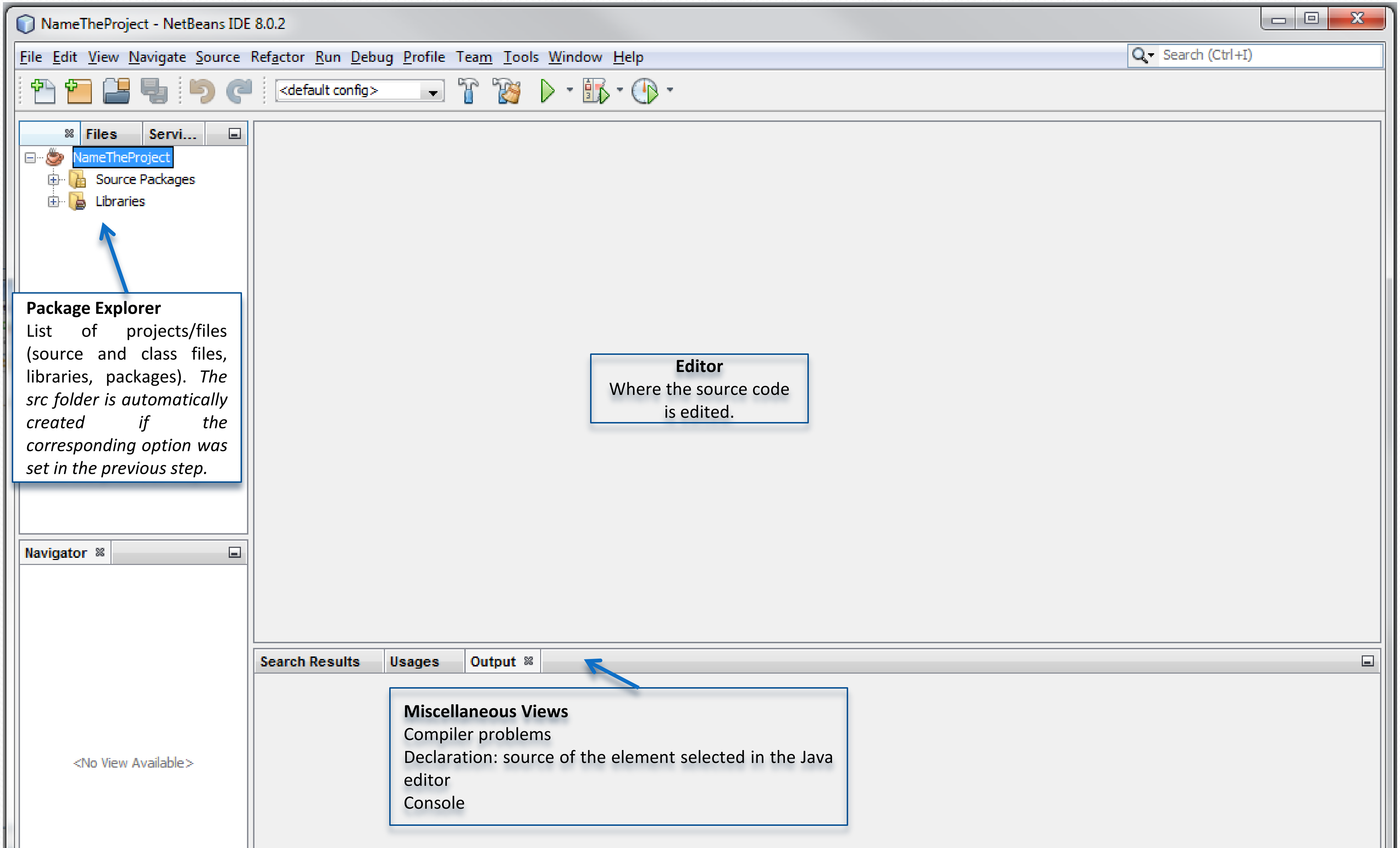
Different users and projects can share the same compilation libraries (see Help for details).

□ Create Main Class nametheproject.NameTheProject

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

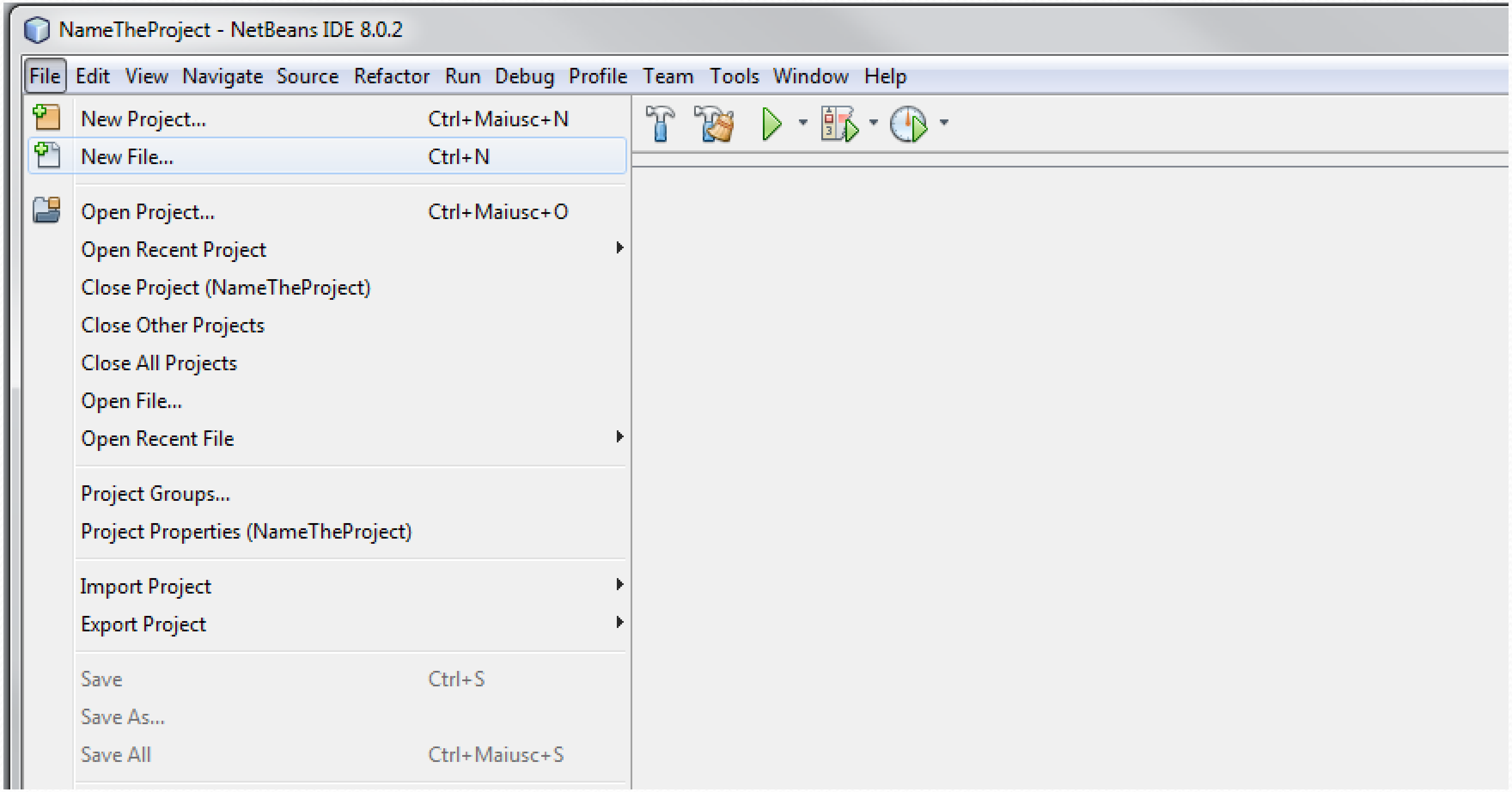


Package Explorer
List of projects/files (source and class files, libraries, packages). *The src folder is automatically created if the corresponding option was set in the previous step.*

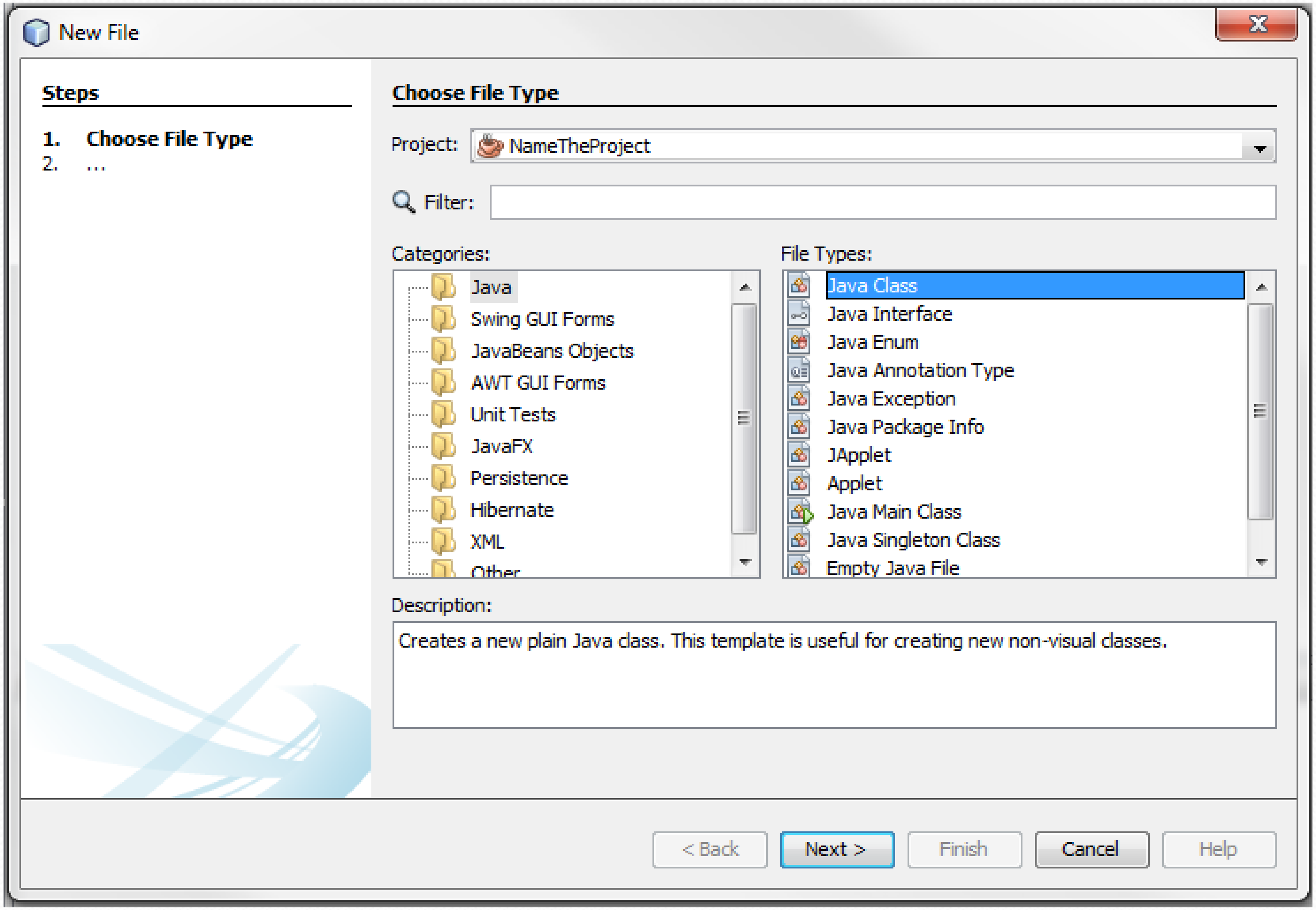
Editor
Where the source code is edited.

Miscellaneous Views
Compiler problems
Declaration: source of the element selected in the Java editor
Console

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



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.



New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project: NameTheProject

Filter:

Categories:

- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- JavaFX
- Persistence
- Hibernate
- XML
- Other

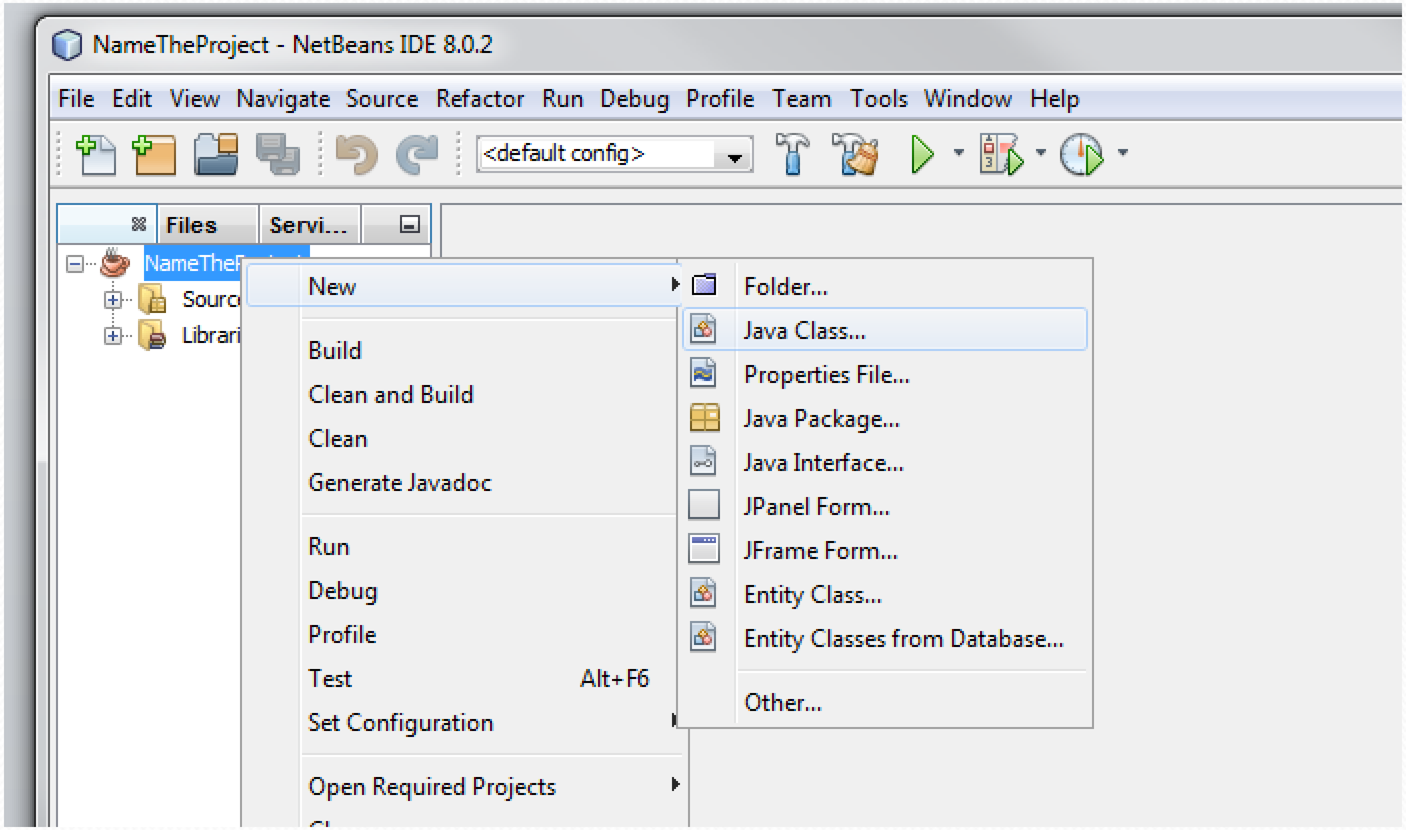
File Types:

- Java Class
- Java Interface
- Java Enum
- Java Annotation Type
- Java Exception
- Java Package Info
- JApplet
- Applet
- Java Main Class
- Java Singleton Class
- Empty Java File

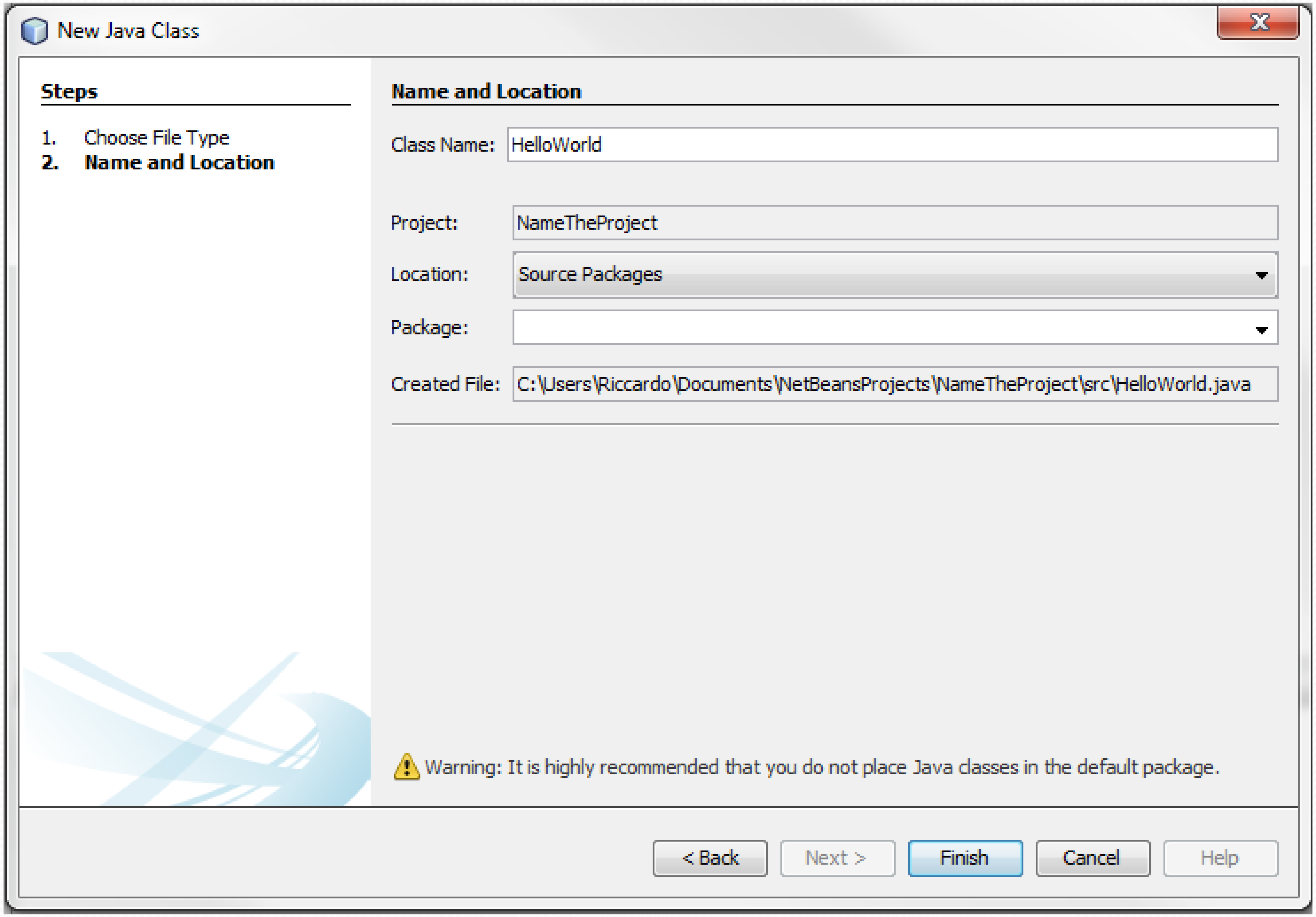
Description:

Creates a new plain Java class. This template is useful for creating new non-visual classes.

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



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



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: HelloWorld

Project: NameTheProject

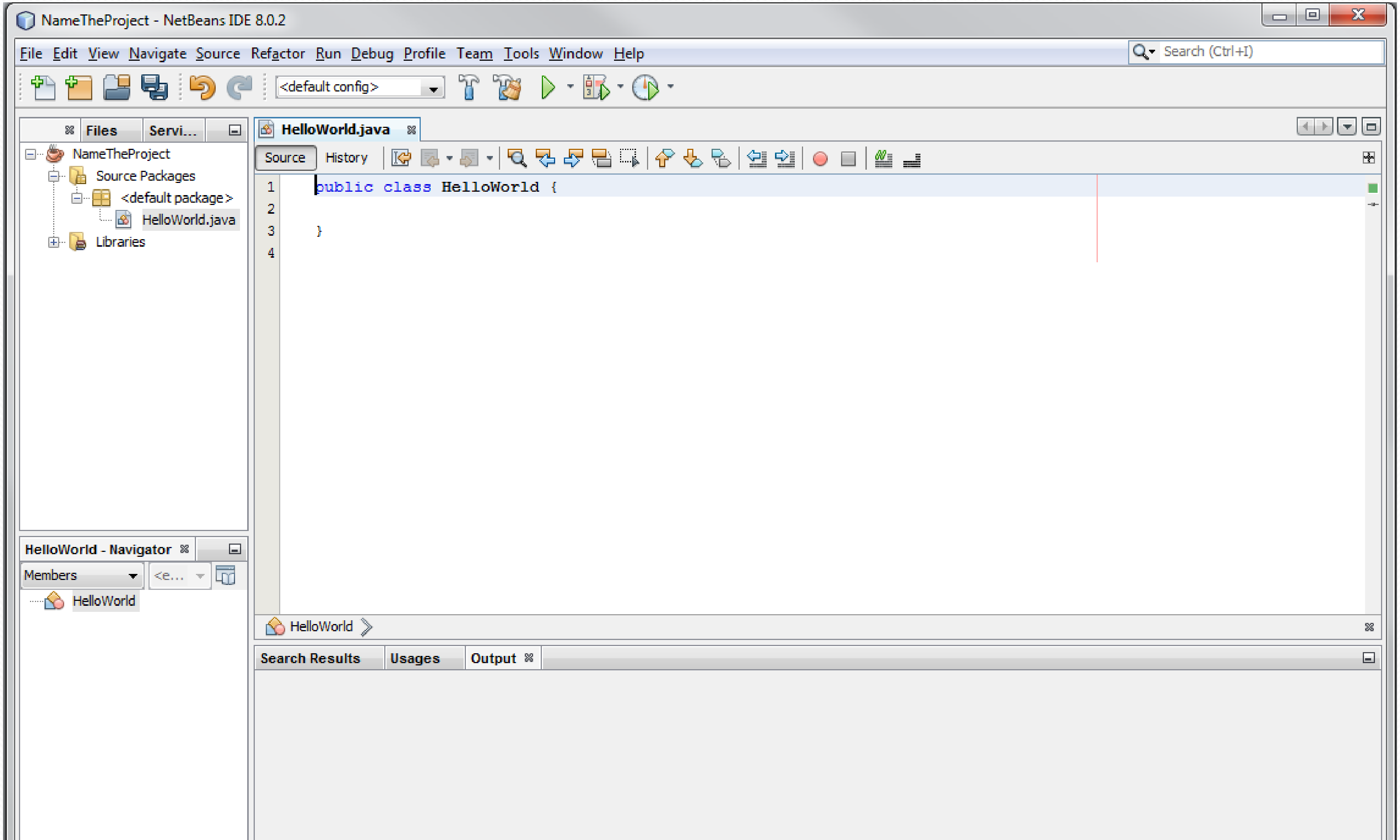
Location: Source Packages

Package:

Created File: C:\Users\Riccardo\Documents\NetBeansProjects\NameTheProject\src\HelloWorld.java

⚠ Warning: It is highly recommended that you do not place Java classes in the default package.

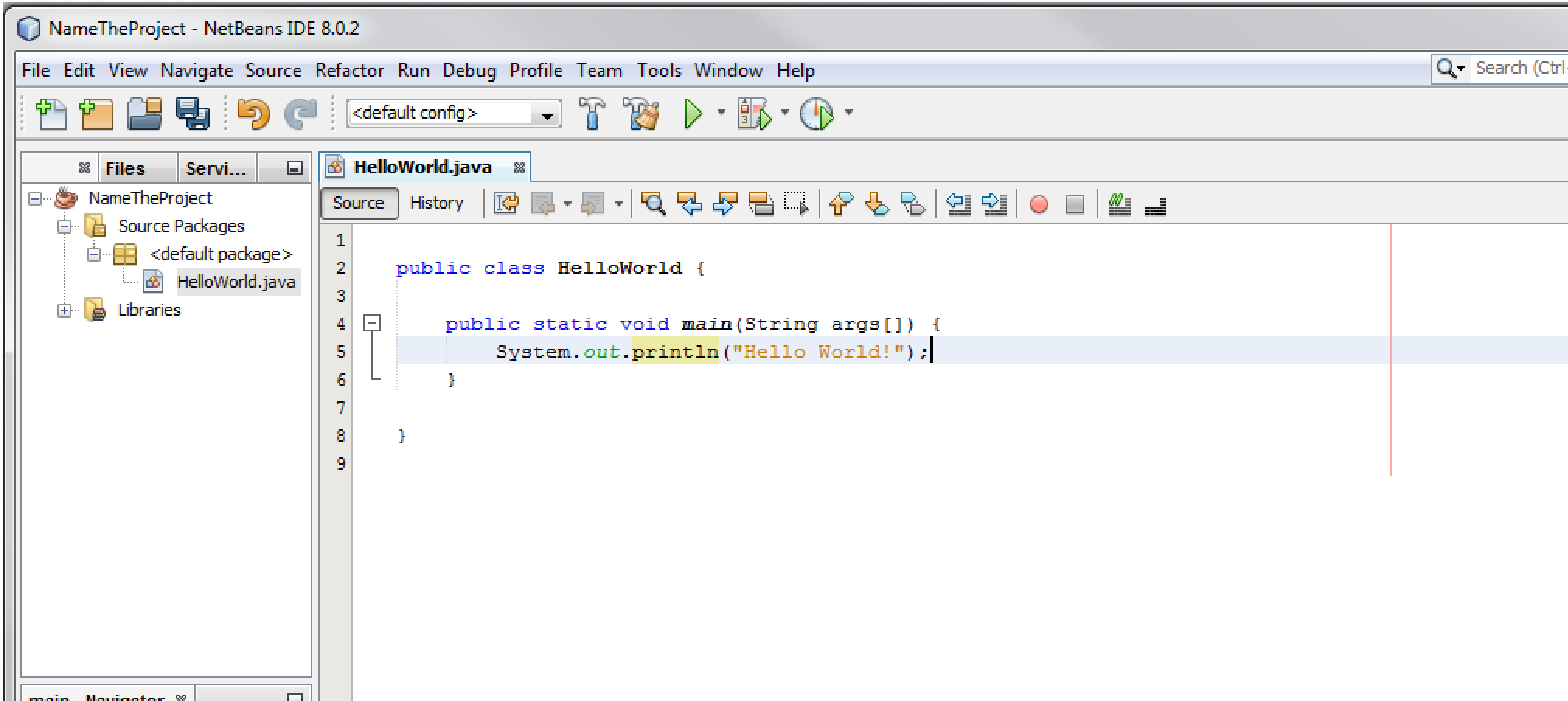
Step9: Now you have the editor space, start coding.



```
public class HelloWorld {

}
```

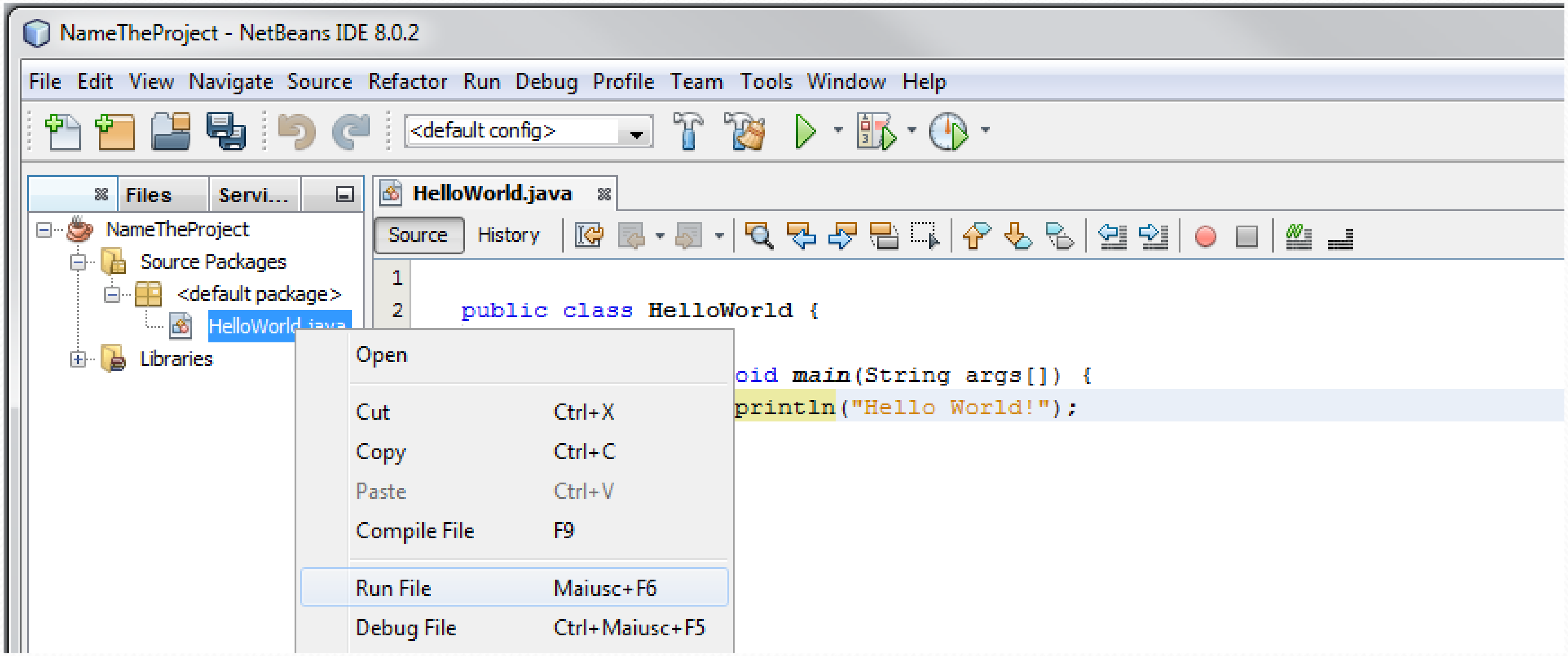
Writing the code

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



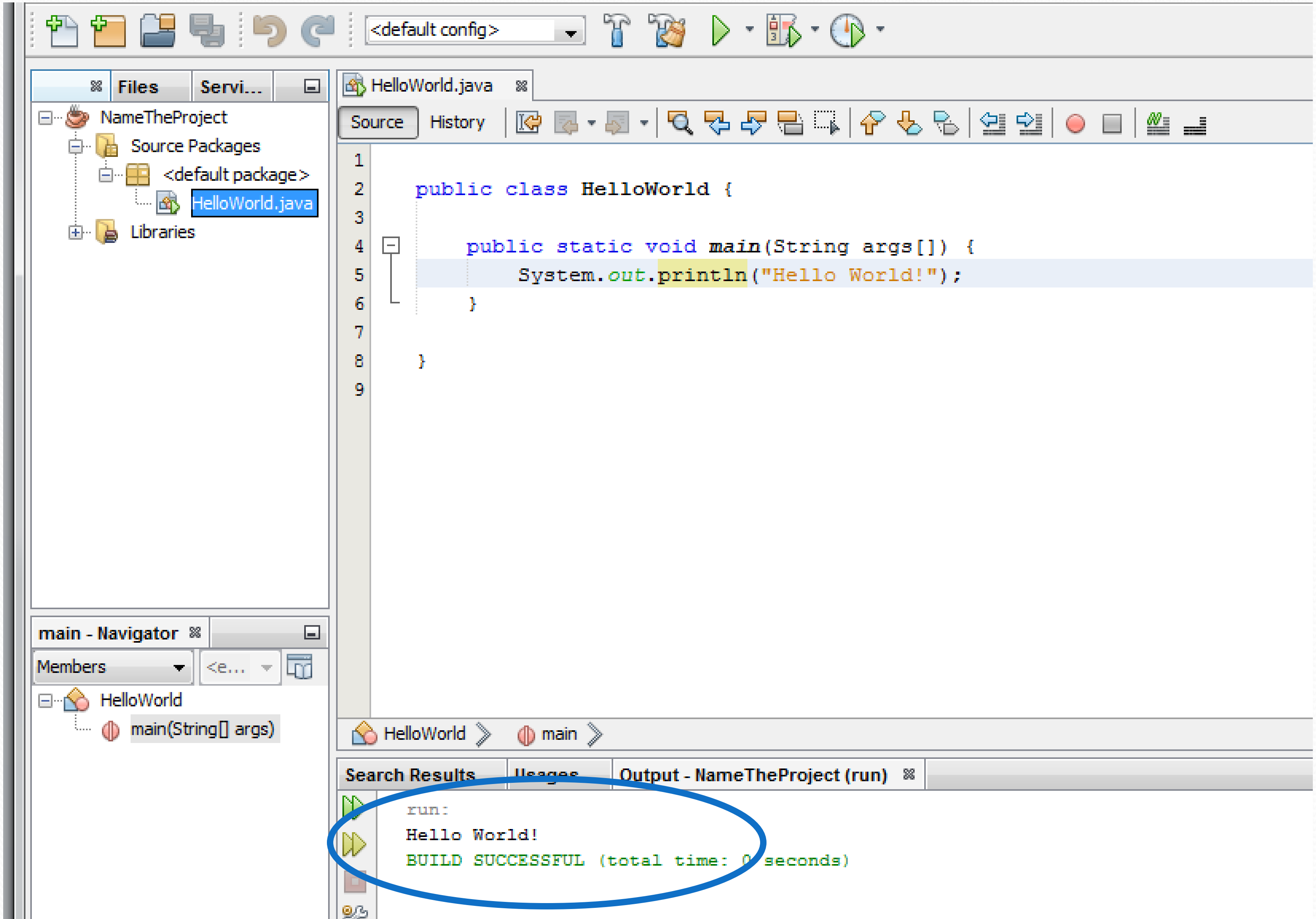
```
public class HelloWorld {

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

}
```

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



```
public class HelloWorld {
oid main(String args[]) {
println("Hello World!");
```

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



```
public class HelloWorld {

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

}
```

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

Creating a .jar file

In NetBeans every time you build the project, the .jar file is updated and saved in the dist/ folder contained in the project main directory.

Executing a jar file

- The basic command is: java -jar *jar-file*
- If the runtime environment has no information about which class within the jar file is the application's entry point (class containing the main method of your application), you must add a Main-Class header to the JAR file's manifest. See: http://docs.oracle.com/javase/tutorial/deployment/jar/appman.html

Some Information and FAQ on NetBeans

Features of NetBeans

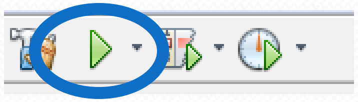
- NetBeans has the basic features required for editing, running, and debugging Java code. In addition to basic programming features, it support for **more advanced Java development tools** such as Ant, CVS, JUnit, and refactoring.

Adding NetBeans plugin

To add plugins you have just click on *Tool* → *Plugins*. A windows will open where you can choice which plugin to download and install.

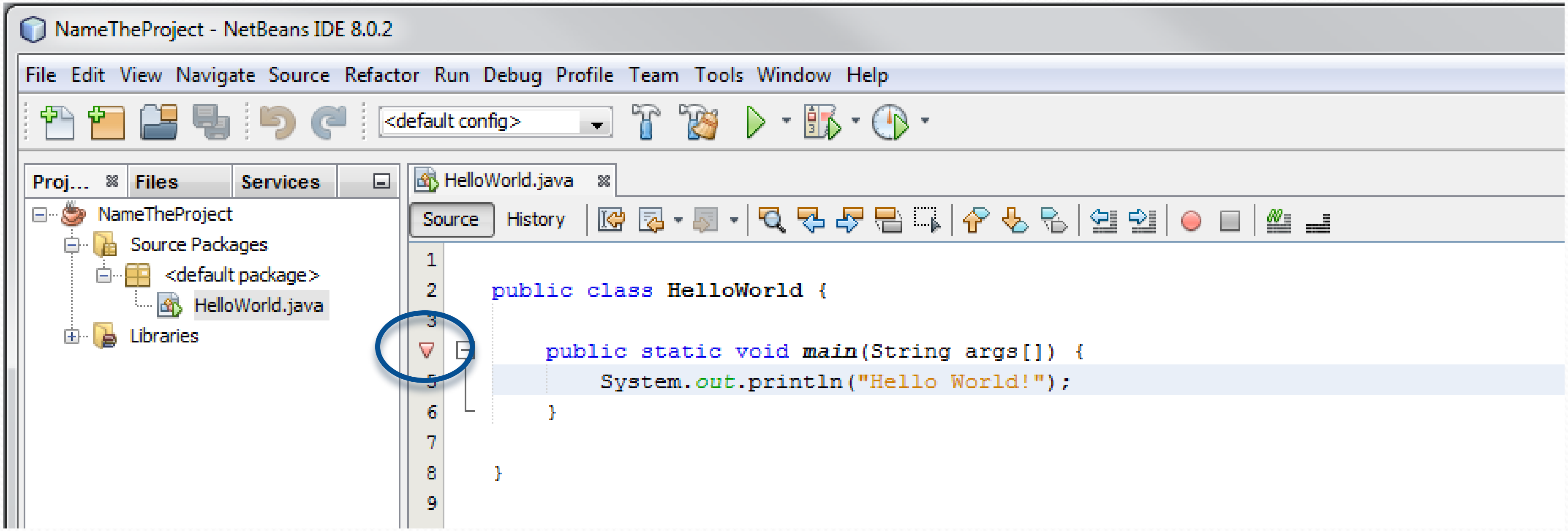
Running code

- 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.



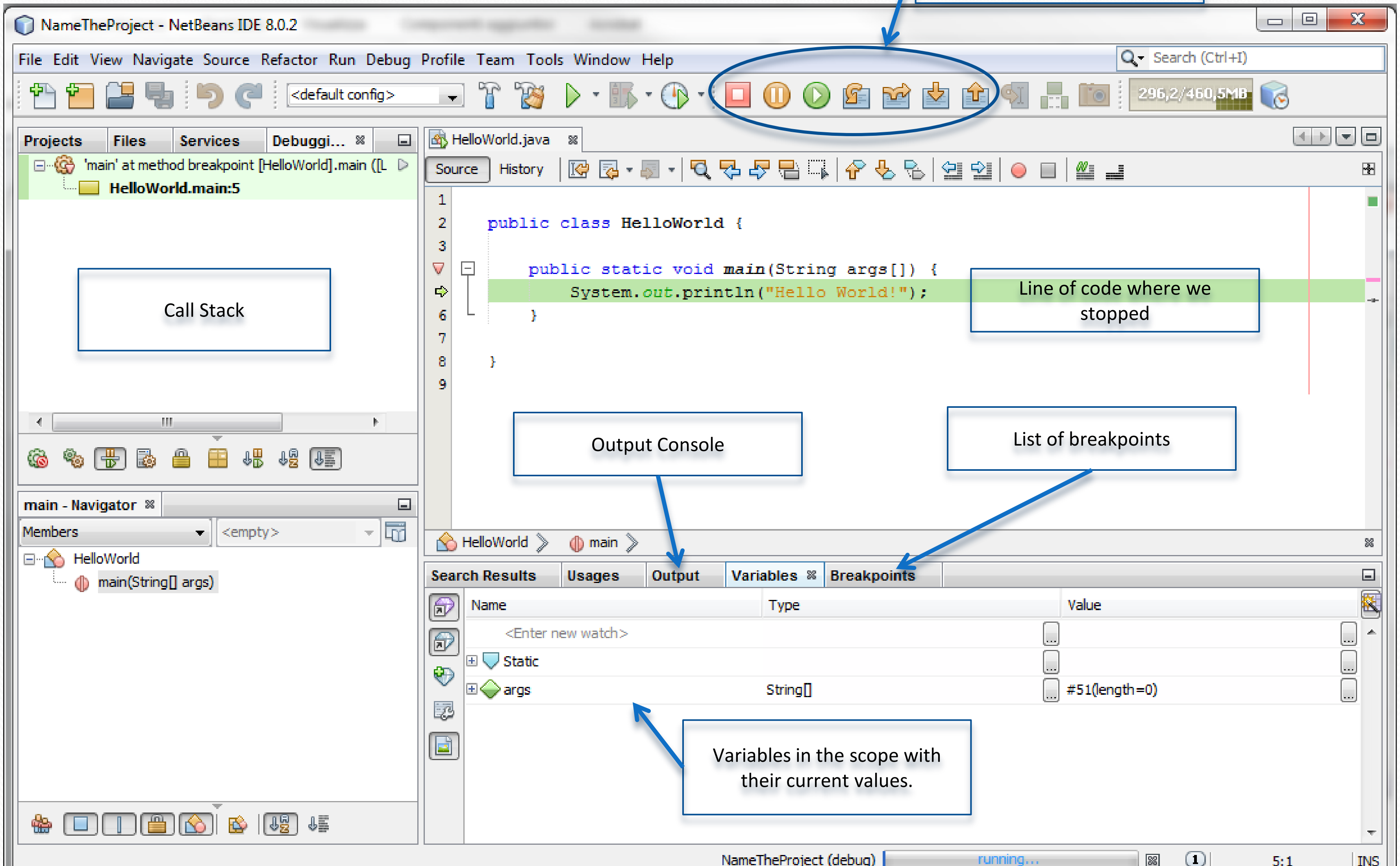
```

public class HelloWorld {

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

}

```

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



NameTheProject - NetBeans IDE 8.0.2

'main' at method breakpoint [HelloWorld].main ([L

HelloWorld.main:5

```
public class HelloWorld {

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

}
```

Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#51(length=0)

Call Stack

Line of code where we stopped

Output Console

List of breakpoints

Variables in the scope with their current values.

The famous Dos and Don’ts:

- Ave 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.