Eclipse IDE Tutorial by Lars Vogel

Contents

What is Eclipse?

Most people know Eclipse as an integrated development environment (IDE) for Java. Today it is the leading development environment for Java with a market share of approximately 65%.

Eclipse is created by an Open Source community and is used in several different areas, e.g. as a development environment for Java or Android applications. Eclipse's roots go back to 2001.

The Eclipse Open Source community has over 200 Open Source projects covering different aspects of software development.

The Eclipse projects are governed by the Eclipse Foundation. The Eclipse Foundation is a non-profit, member supported corporation that hosts the Eclipse Open Source projects and helps to cultivate both an Open Source community and an ecosystem of complementary products and services.

The Eclipse IDE can be extended with additional software components. Eclipse calls these software components plug-ins. Several Open Source projects and companies have extended the Eclipse IDE.

It is also possible to use Eclipse as a basis for creating general purpose applications. These applications are known as Eclipse Rich Client Platform (Eclipse RCP) applications.

Eclipse Public License

The Eclipse Public License (EPL) is an Open Source software license used by the Eclipse Foundation for its software. The EPL is designed to be business-friendly. EPL licensed programs can be used, modified, copied and distributed free of charge and receiver of EPL-licensed software can choose to use this software in closed source programs. Only modifications in the original EPL code must be released.

The Eclipse Foundation also validates that source code contributed to Eclipse projects is free of Intellectual property (IP) issues. This process is known as IP cleansing.

The permissive EPL and the IP cleansing effort of the Eclipse Foundation makes reusing the source code of Eclipse projects attractive

Eclipse Installation

Java Requirements of Eclipse

Eclipse requires an installed Java Runtime. Eclipse 4.2 requires at least Java 5 to run.

For this tutorial you should use Java in version 6 or higher.

The Eclipse IDE contains its own Java compiler. The Java Development Tools are required if you compile Java source code outside Eclipse and for advanced development scenarios. For example if you use automatic builds or if you develop web development

Installation of Java

Java might already be installed on your machine. You can test this by opening a console (if you are using Windows: Win+R, enter cmd and press Enter) and by typing in the following command:

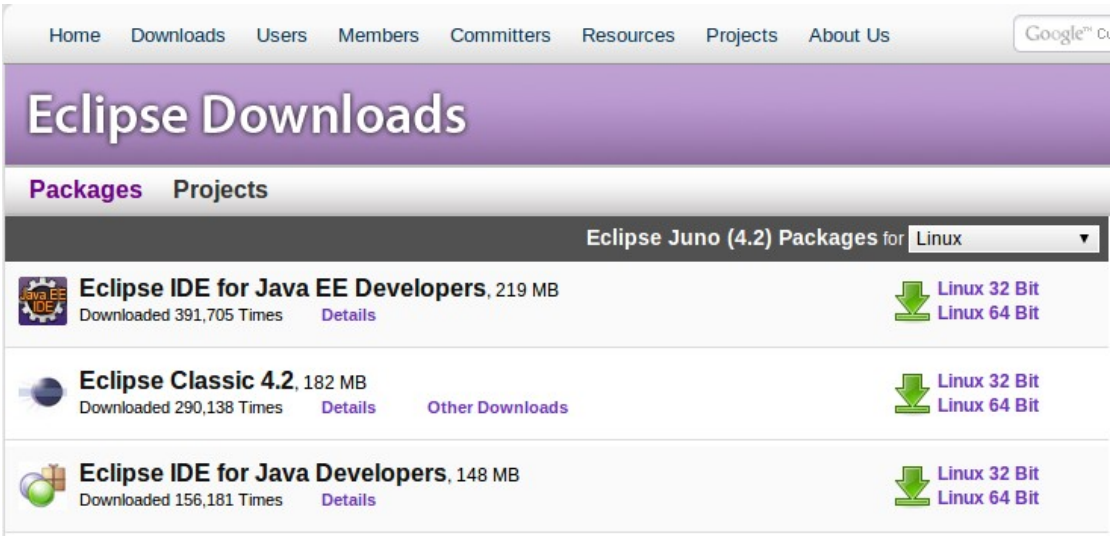
```
java -version
```

If Java is correctly installed, you should see some information about your Java installation. If the command line returns the information that the program could not be found, you have to install Java.

A Google search for How to install JDK on YOUR_OS should result in helpful links. Replace YOUR_OS with your operating system, e.g. Windows, Ubuntu, Mac OS X, etc.

Download Eclipse

The following screenshot shows the Eclipse download website for a Linux system, press on the link beside the package, e,g, Linux 64 Bit to start the download.



The download is a .zip file

Install Eclipse

After you downloaded the .zip file which contains the Eclipse distribution you unpack it to a local directory.

Most operating system can extract zip files in their file browser, e.g. Windows7 via right mouse click on the file and selecting "Extract all...". If

in doubt, search via Google for "How to unzip a file on ...", again replacing "..." with your operating system.

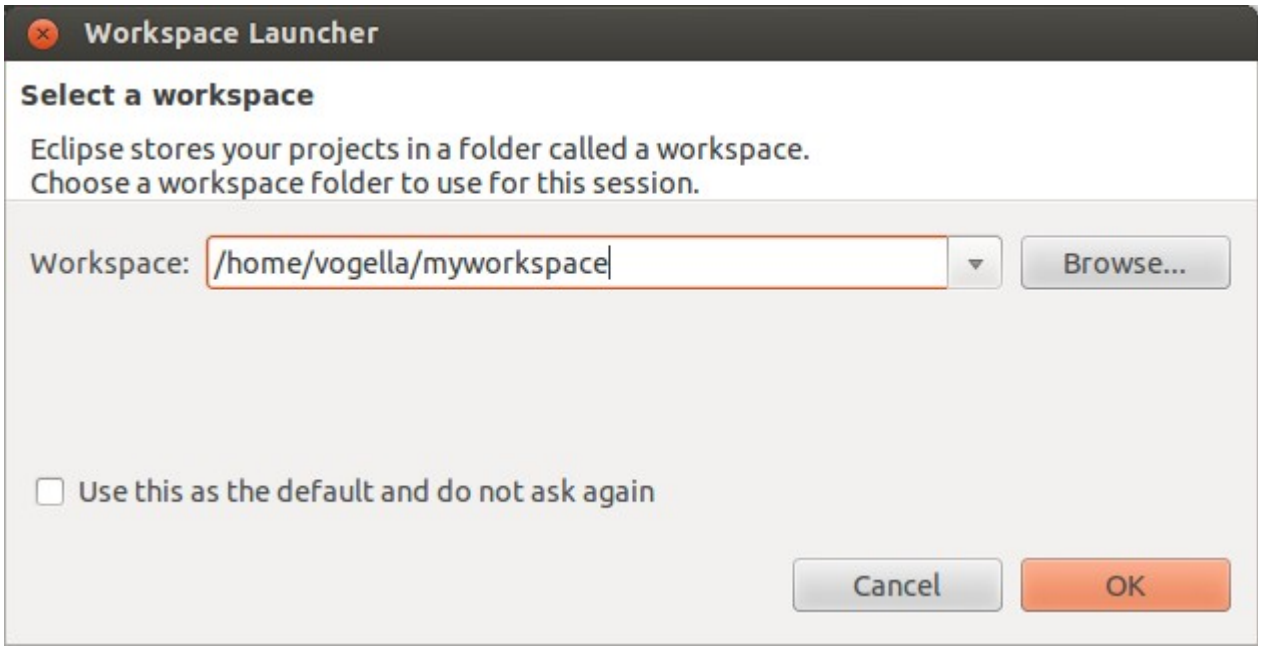
Use a directory path which does not contain spaces in its name, as Eclipse sometimes has problems with that.

After unpacking the downloaded zip file, Eclipse is ready to be used; no additional installation procedure is required.

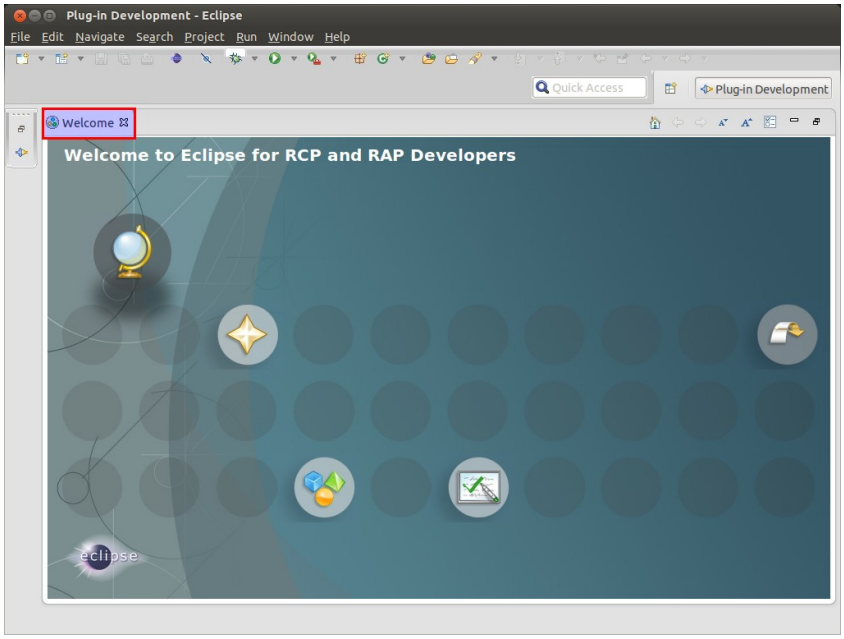
Getting started

To start Eclipse double-click on the file eclipse.exe (Microsoft Windows) or eclipse (Linux / Mac) in the directory where you unpacked Eclipse.

The system will prompt you for a *workspace*. The *workspace* is the place in which you work. Select an empty directory and press the *OK* button.



Eclipse will start and show the Welcome page. Close the welcome page by pressing the *X* beside*Welcome*.



Appearance

The appearance of Eclipse can be changed. By default Eclipse ships with a few themes but you can also extend Eclipse with new themes.

To change the appearance, select from the menu *Window* → *Preferences* → *General* → *Appearance*

The *Theme* selection allows you to change the appearance of your Eclipse IDE. Please note that you need to restart Eclipse to apply a new styling correctly

Eclipse user interface overview

Eclipse provides *Perspectives*, *Views* and *Editors*. *Views* and *Editors* are grouped into *Perspectives*.

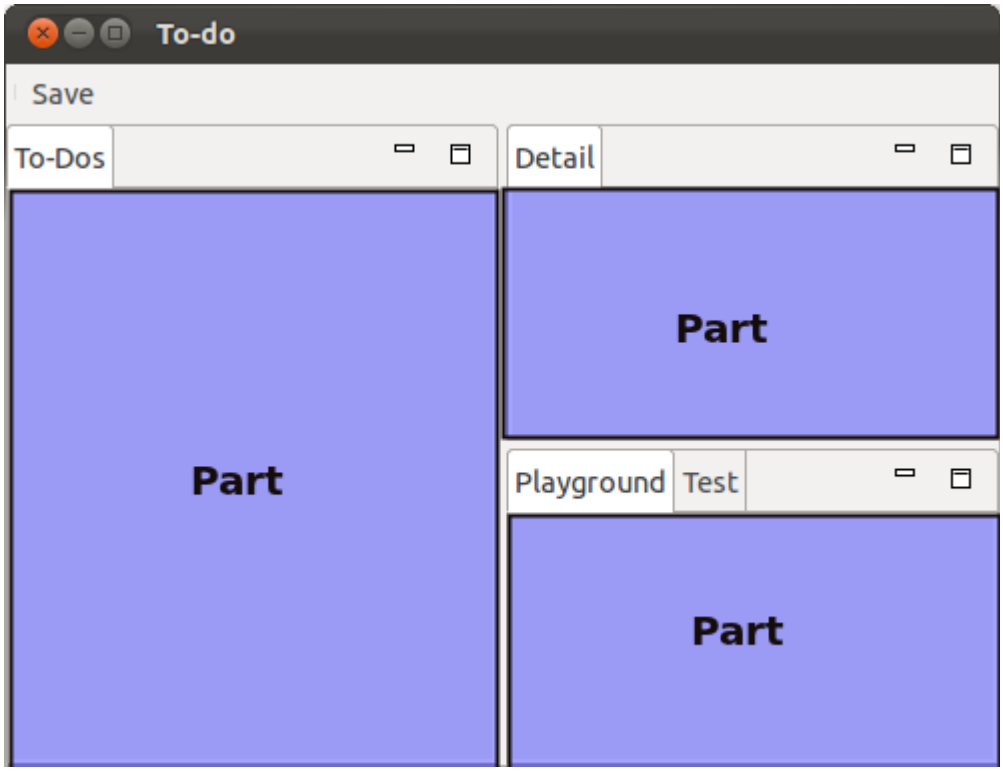
Workspace

The *workspace* is the physical location (file path) you are working in. Your projects, source files, images and other artifacts can be stored and saved in your workspace but you can also refer to external resources, e.g. projects, in your *workspace*.

You can choose the workspace during startup of Eclipse or via the menu (*File* → *Switch Workspace* →*Others*) .

Parts

Parts are user interface components which allow you to navigate and modify data. *Parts* are typically divided into *Views* and *Editors*.



The distinction into *Views* and *Editors* is primarily not based on technical differences, but on a different concept of using and arranging these *Parts*.

A *View* is typically used to work on a set of data, which might be a hierarchical structure. If data is changed via the *View*, this change is typically directly applied to the underlying data structure.
A *View* sometimes allows us to open an *Editor* for a selected set of the data.

An example for a *View* is the *Java Package Explorer*, which allow you browse the files of Eclipse Projects. If you choose to change data in the Package Explorer, e.g. if you rename a file, the file name is directly changed on the file system.

Editors are typically used to modify a single data element, e.g. a file or a data object. To apply the changes made in an editor to the data structure, the user has to explicitly save the editor content.

Editors were traditionally placed in a certain area, called the *editor area*. Until Eclipse 4 this was a hard limitation, it was not possible to move an *Editor* out of this area; Eclipse 4 allows the user to place *Editors*at any position in a *Perspective* or even outside a *Perspective*.

For example the Java Editor is used to modify Java source files. Changes to the source file are applied once the user selects the *Save* command.

Perspective

A *Perspective* is a visual container for a set of *Parts*. The Eclipse IDE uses *Perspectives* to arrange *Parts*for different development tasks.

You can change the layout and content within a *Perspective* by opening or closing *Parts* and by re-arranging them.

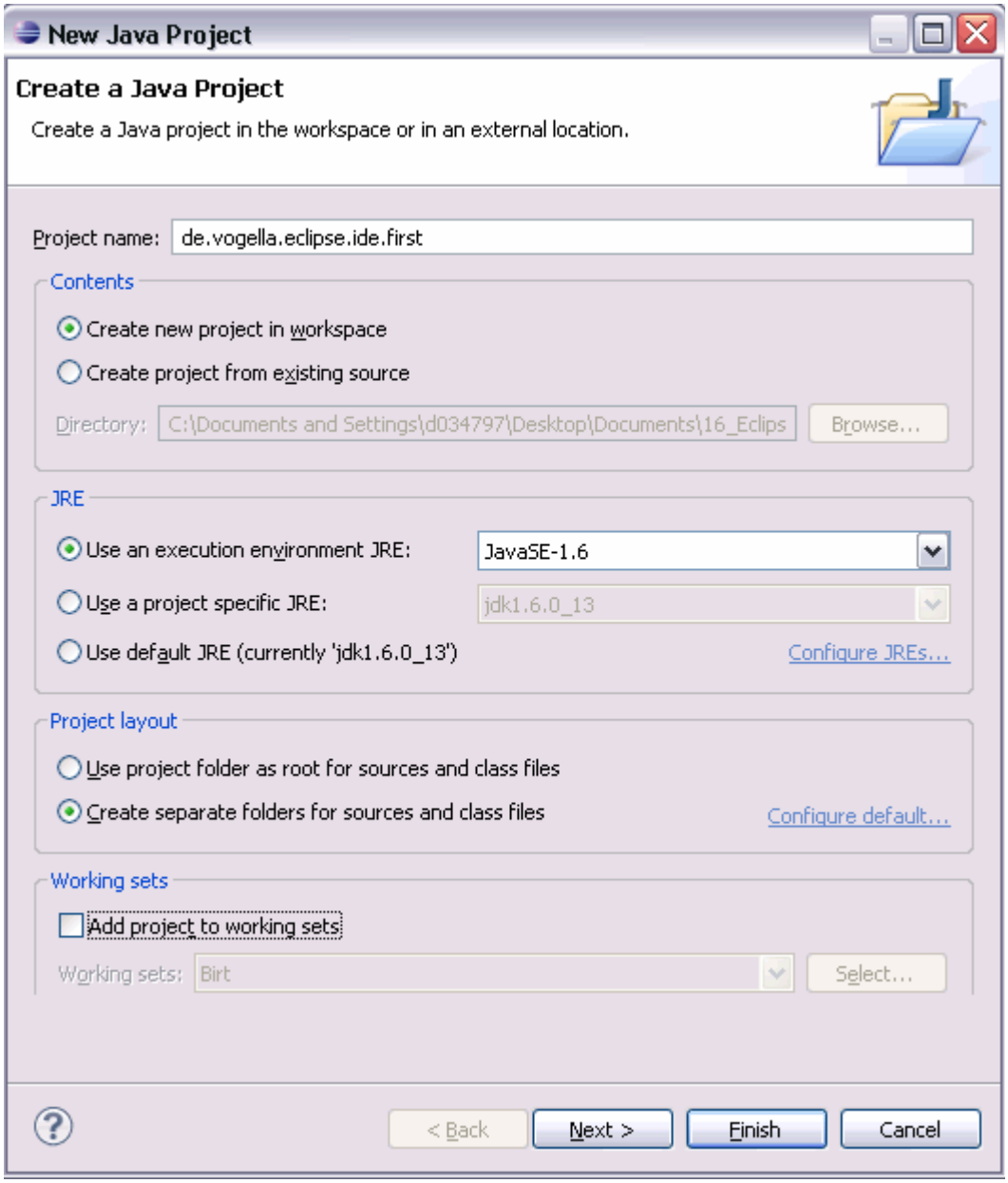
As of Eclipse 4 *Perspectives* are optional elements for Eclipse applications.

Create your first Java program

The following describes how to create a minimal Java program using Eclipse. It is tradition in the programming world to create a small program which writes "Hello World" to the console. We will adapt this tradition and will write "Hello Eclipse!" to the console.

Create project

Select from the menu *File* → *New* → *Java project*.
Enter de.vogella.eclipse.ide.first as the project name. Select the *Create separate folders for sources and class files* flag.

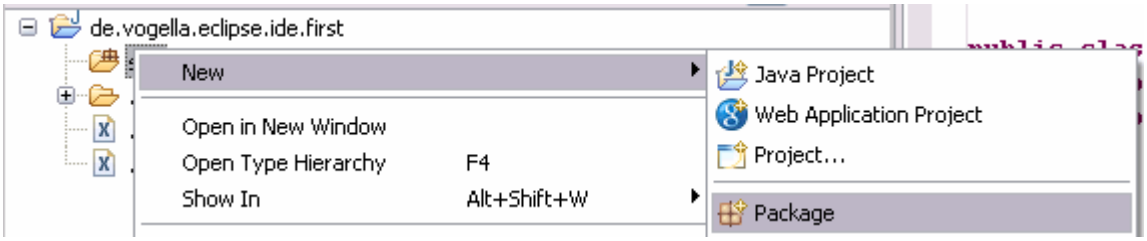


Press the *Finish* button to create the project. A new project is created and displayed as a folder. Open thede.vogella.eclipse.ide.first folder and explore the content of this folder

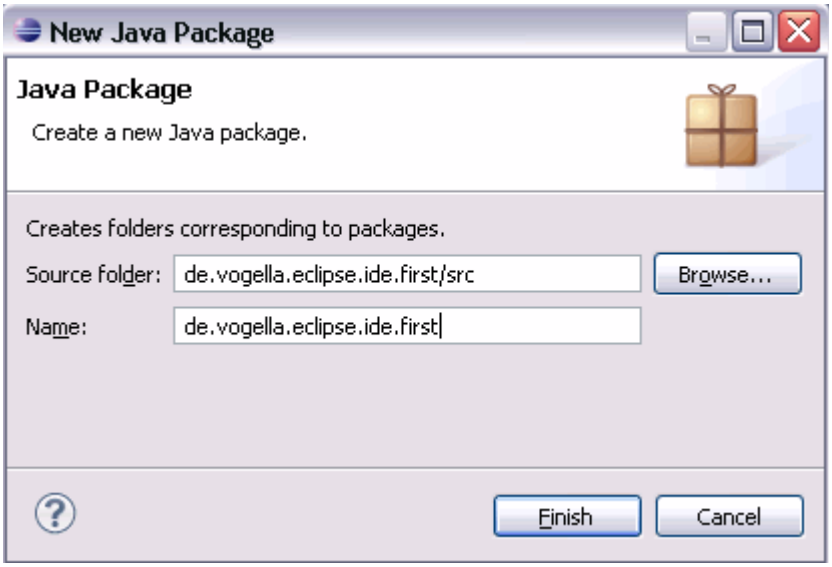
Create package

In the following step you will create a new package. A good convention is to use the same name for the top level package and the project.

To create the de.vogella.eclipse.ide.first package, select the folder src, right click on it and select *New* → *Package*.

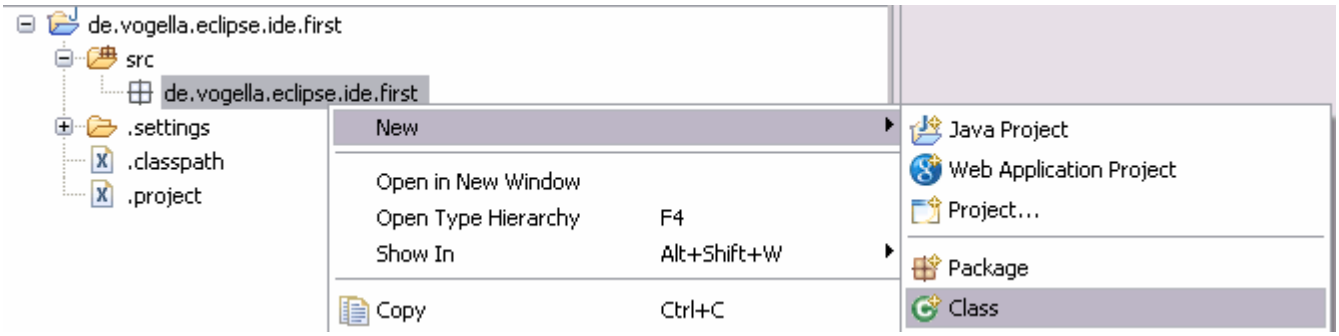


Enter the name of your new package in the dialog and press the *Finish* button.

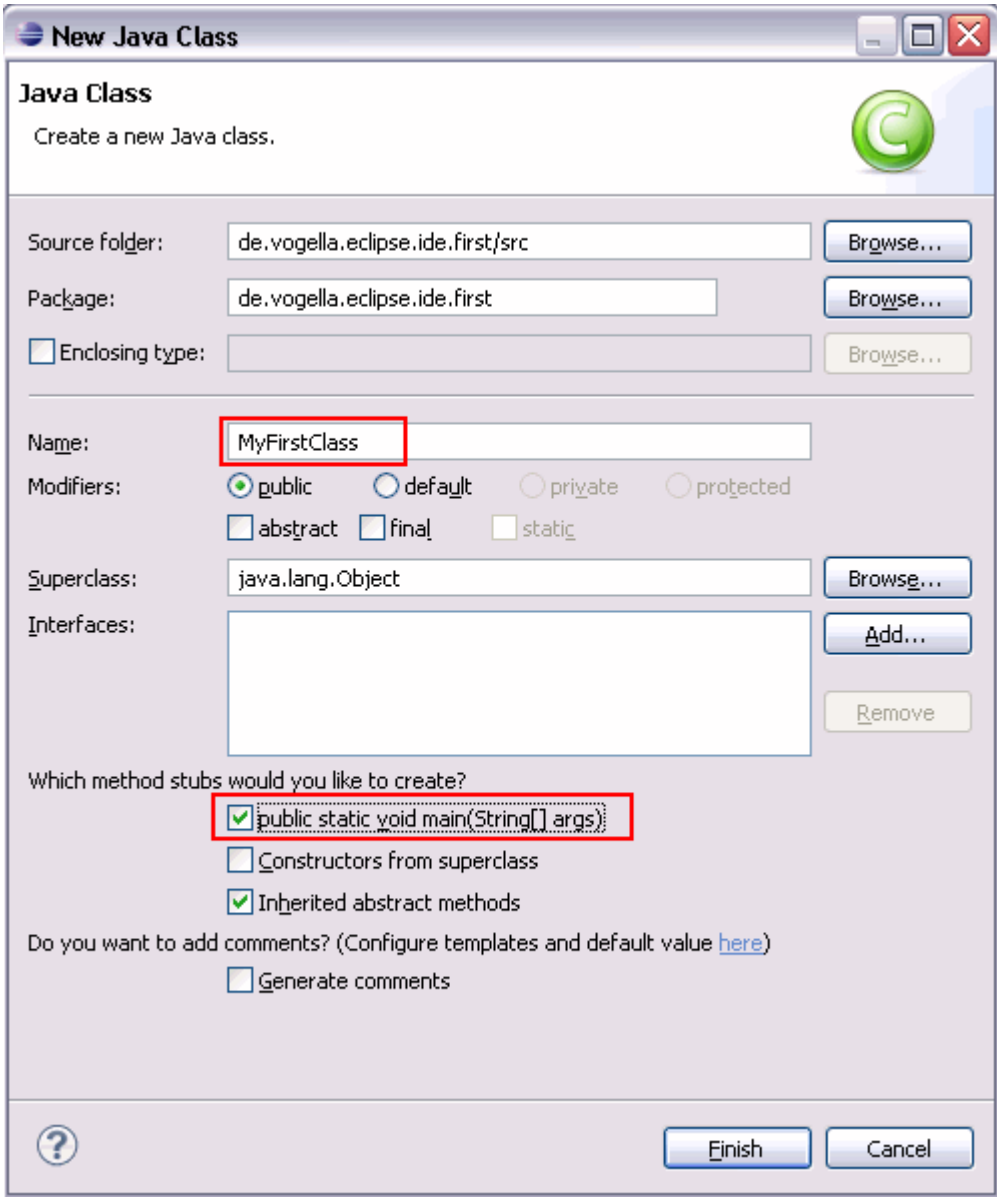


Create Java class

Create a Java class. Right click on your package and select *New → Class*.



Enter MyFirstClass as the class name and select the *public static void main (String[] args)* flag.



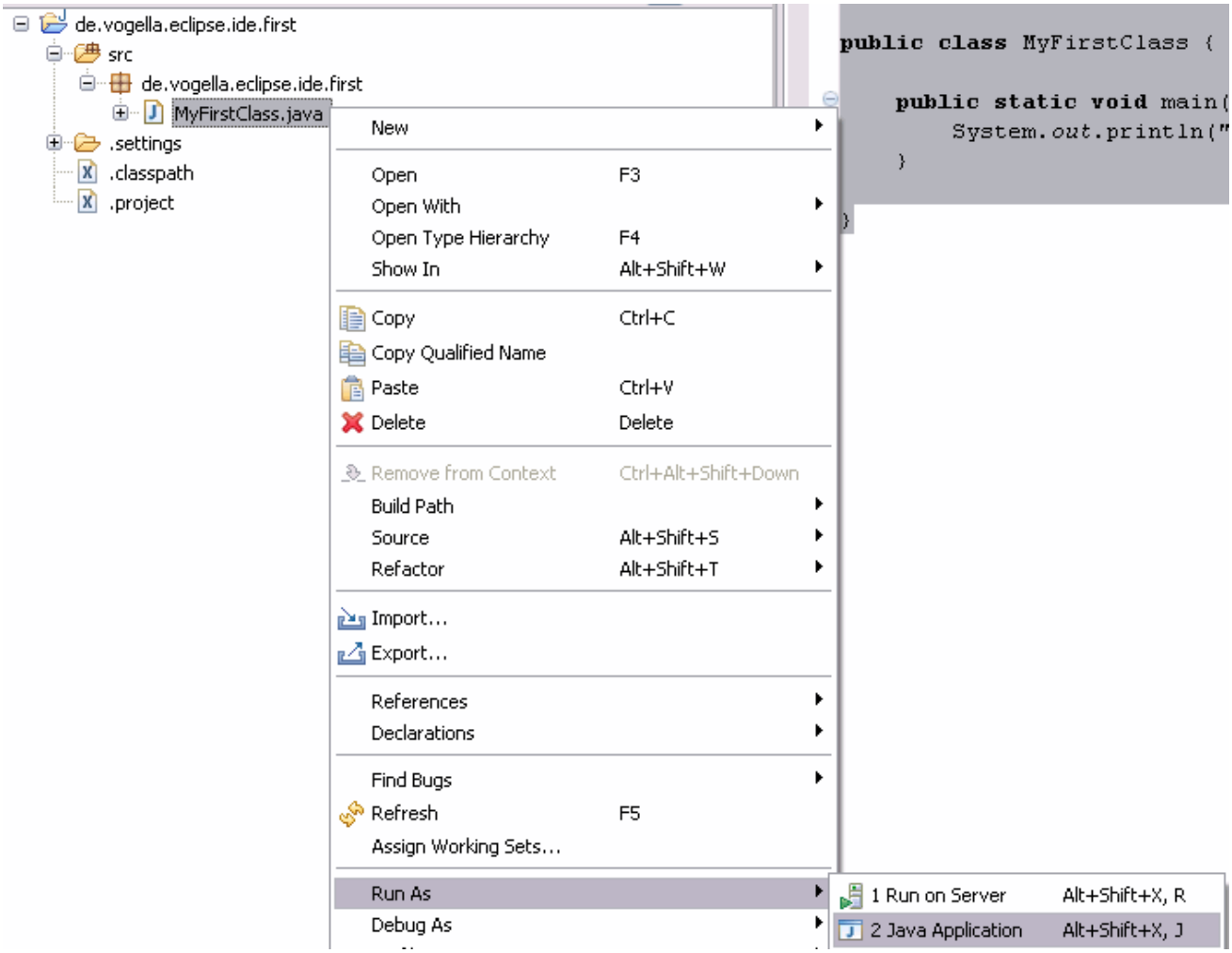
Press the *Finish* button.

This creates a new file and opens the *Editor* for Java source files. Change the class to the following example.

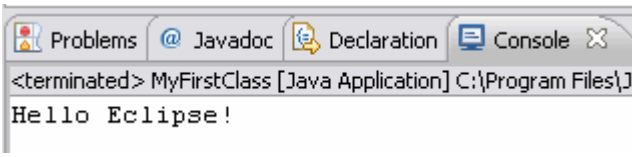
```
package de.vogella.eclipse.ide.first;


public class MyFirstClass {

  public static void main(String[] args) {

    System.out.println("Hello Eclipse!");

  }

}
```

Run your project in Eclipse

Now run your code. Right click on your Java class and select *Run-as → Java application*.



Eclipse will run your Java program. You should see the output in the *Console View*.



Congratulations! You created your first Java project, a package, a Java class and you ran this program inside Eclipse.