

Eclipse IDE Tutorial

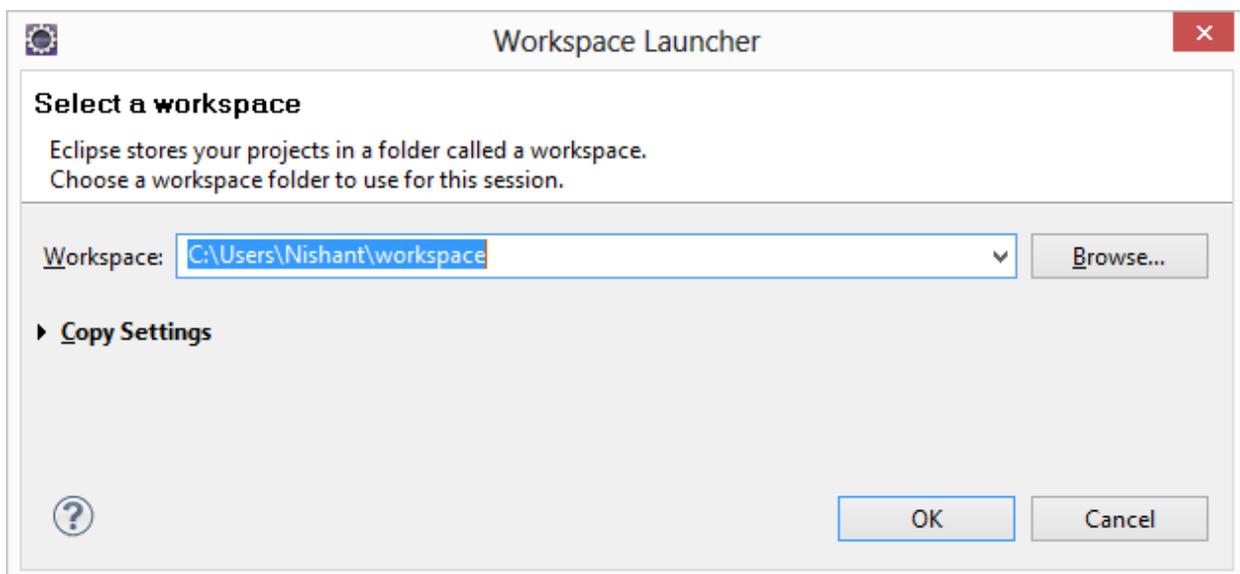
To start Eclipse, double-click on the eclipse.exe (Microsoft Windows) or eclipse (Linux / Mac) file in the directory where you unpacked Eclipse.

The system will prompt you for a *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. The *workspace* also contains preferences settings, plug-in specific metadata, logs etc.

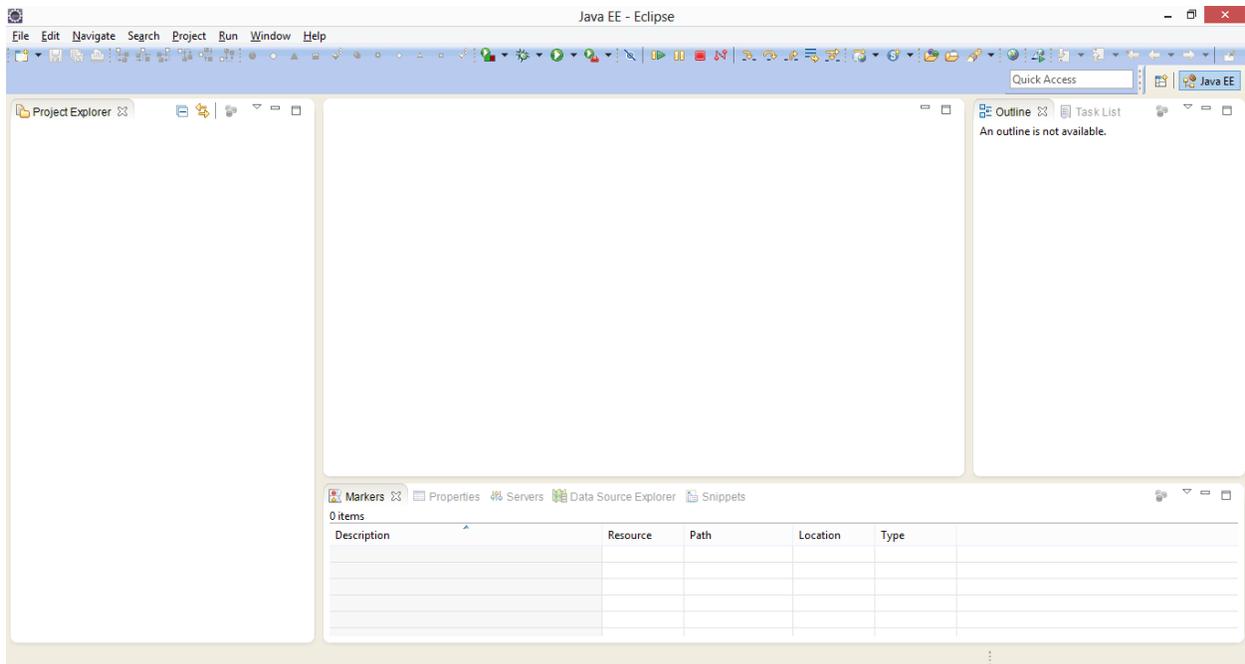
You typically use different *workspaces* if you require different settings for your project or if you want to divide your projects into separate directories.

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

Select an empty directory and click the *OK* button.



Once you select the workspace, the application will look similar to the following screenshot.



Eclipse projects

An Eclipse project contains source, configuration and binary files related to a certain task and groups them into buildable and reusable units. An Eclipse project can have *natures* assigned to it which describe the purpose of this project. For example, the Java *nature* defines a project as Java project. Projects can have multiple natures combined to model different technical aspects.

Projects in Eclipse cannot contain other projects.

Views and editors - parts

Parts are user interface components which allow you to navigate and modify data. A part can have a dropdown menu, context menus and a toolbar.

Parts can be freely positioned in the user interface.

Parts are typically classified into *views* and *editors*. The distinction into views and editors is 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 data.

An example for a view is the *Project Explorer*, which allows you to browse the files of Eclipse projects. If you change data in the *Project Explorer*, e.g., renaming a file, the file name is directly changed on the file system.

Editors are typically used to modify a single data element, e.g., the content of 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.

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. A dirty editor tab is marked with an asterisk to the left of the modified name of the file.

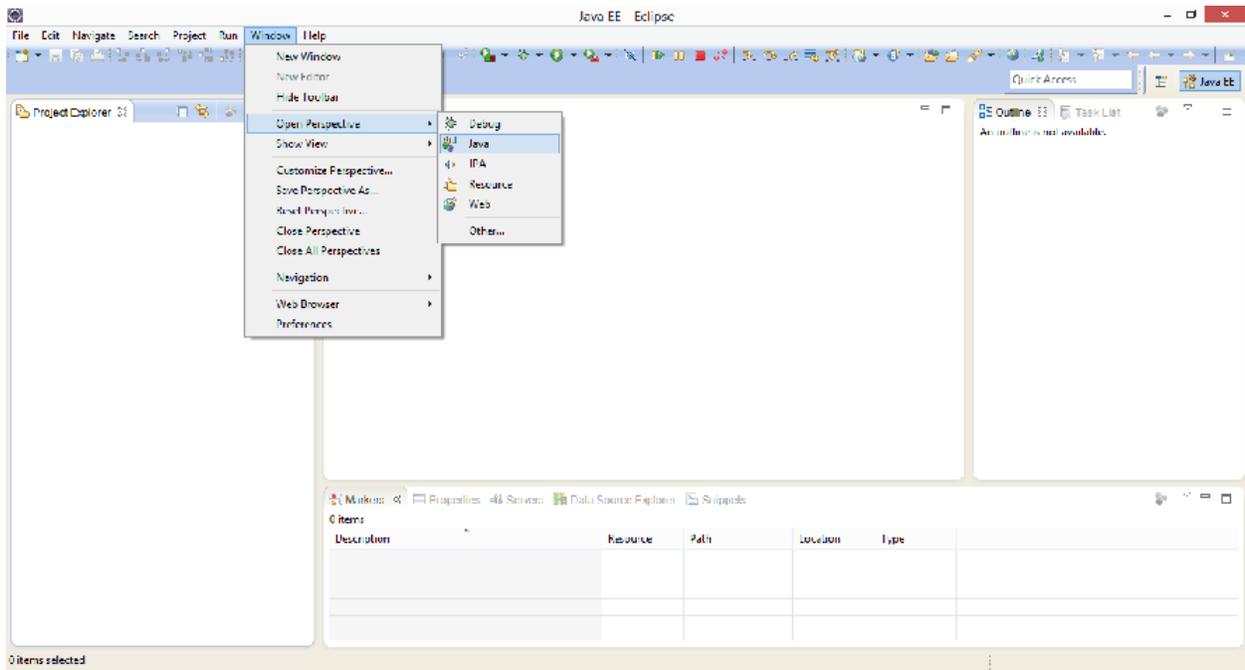
Perspective

A *perspective* is a visual container for a set of parts. Perspectives can be used to store different arrangements of parts. For example, the Eclipse IDE uses them to layout the views appropriate to the task (development, debugging, review, ...) the developer wants to perform.

Open editors are typically shared between perspectives, i.e., if you have an editor open in the *Java* perspective for a certain class and switch to the *Debug* perspective, this *editor* stays open.

You can switch *Perspectives* via the *Window* → *Open Perspective* → *Other...* menu entry.

The main perspectives used for Java development are the *Java* perspective and the *Debug* perspective.



You can change the layout and content within a perspective by opening or closing parts and by re-arranging them.

To open a new part in your current perspective, use the *Window* → *Show View* → *Other...* menu entry.

Perspectives in Eclipse

Eclipse provides different *perspectives* for different tasks. The available *perspectives* depend on your installation.

For Java development you usually use the *Java Perspective*, but Eclipse has much more predefined *perspectives*, e.g., the *Debug perspective*.

Eclipse allows you to switch to another *perspective* via the *Window* → *Open Perspective* → *Other...* menu entry.

Java perspective and Package Explorer

The default *perspective* for Java development can be opened via *Window* → *Open Perspective* → *Java*.

On the left hand side, this perspective shows the *Project Explorer* view, which allows you to browse your *projects* and to select the components you want to open in an editor via a double-click.

For example, to open a Java source file, open the tree under *src*, select the corresponding *.java* file and double-click it. This will open the file in the default Java *editor*.

The following picture shows the Eclipse IDE in its standard Java *perspective*. The *Project Explorer* view is on the left. In the middle you see the open *editors*. Several *editors* are stacked in the same container and you can switch between them by clicking on the corresponding tab. Via drag and drop you can move an editor to a new position in the Eclipse IDE.

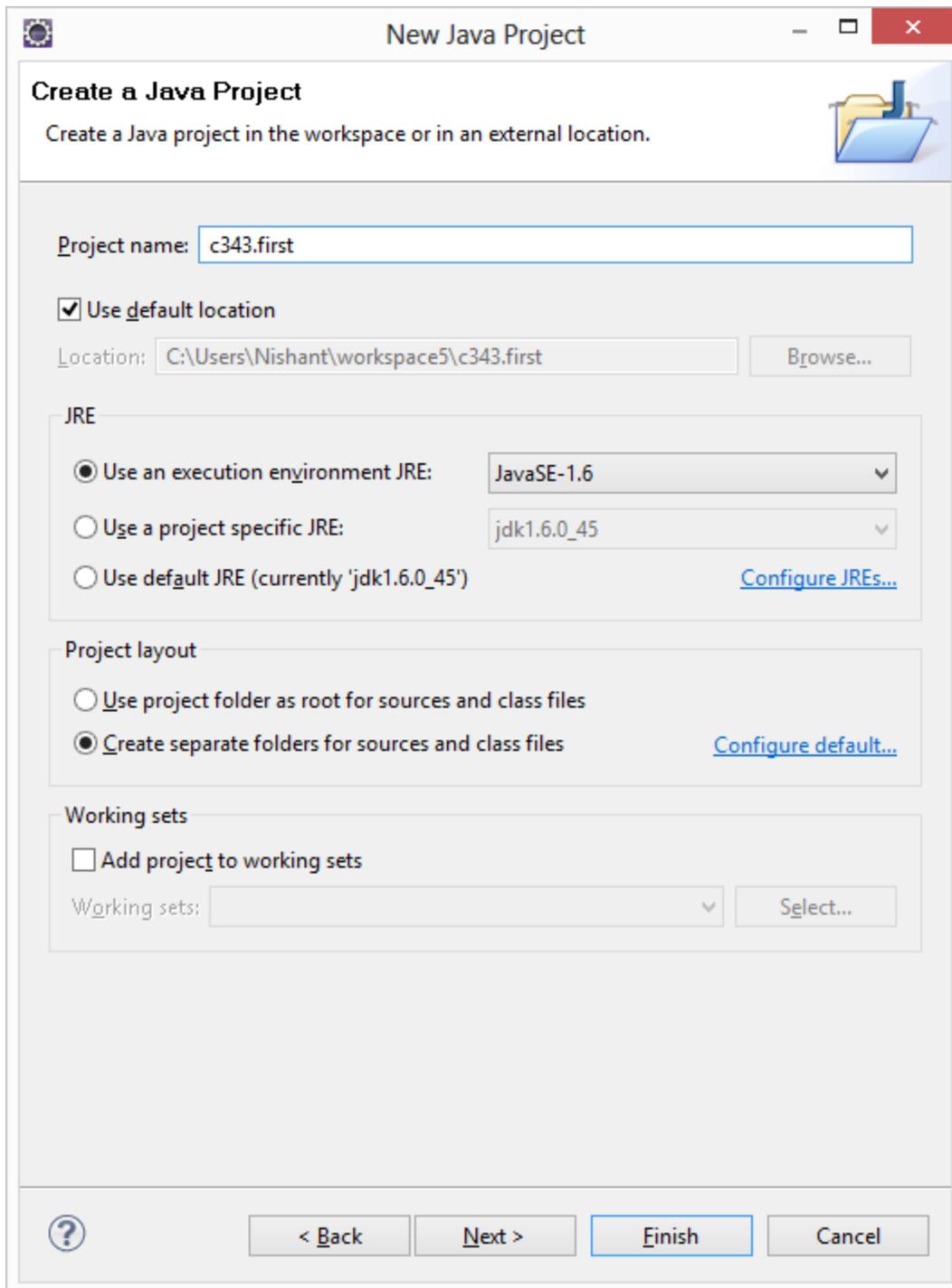
To the right and below the editor area you find more *views* which were considered useful by the developer of the perspective. For example, the *Javadoc* view shows the Javadoc of the selected class or method.

Create your first Java program

Create project

Select *File* → *New* → *Java project* from the menu.

Enter *c343.first* as the project name. Select the *Create separate folders for sources and class files* flag.

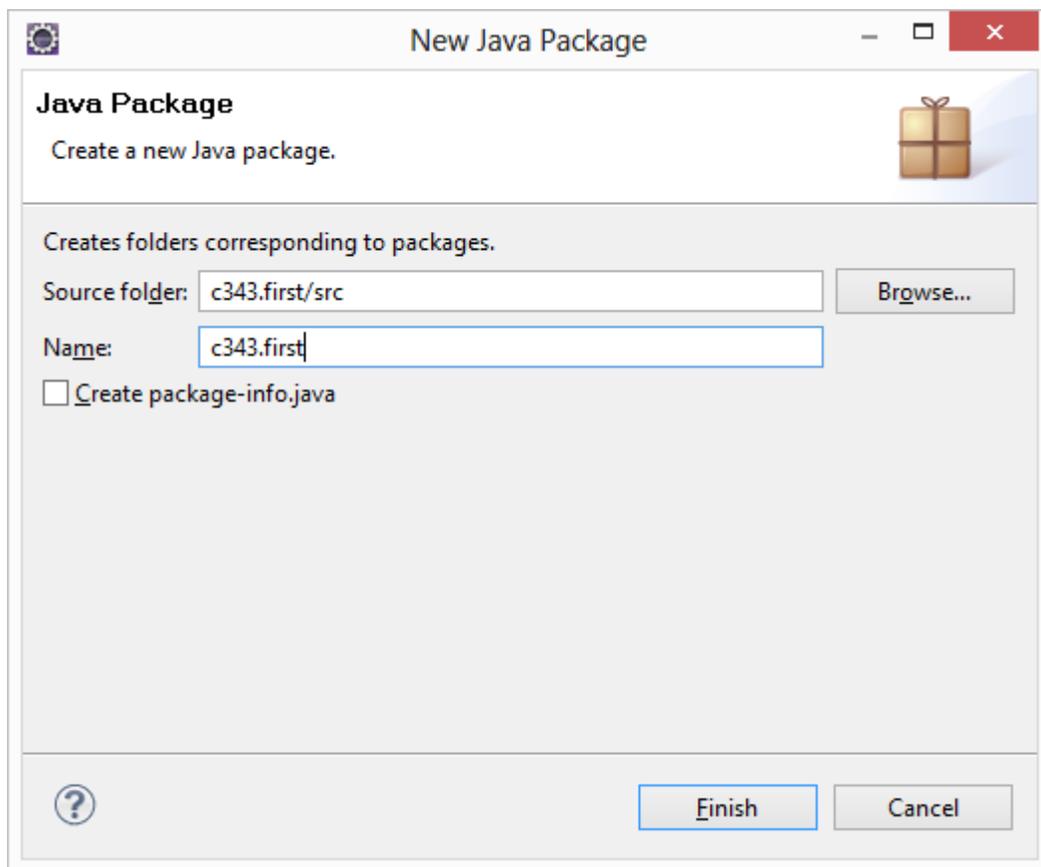


Press the *Finish* button to create the project. A new project is created and displayed as a folder. Open the `c343.first` folder and explore the content of this folder.

Create package

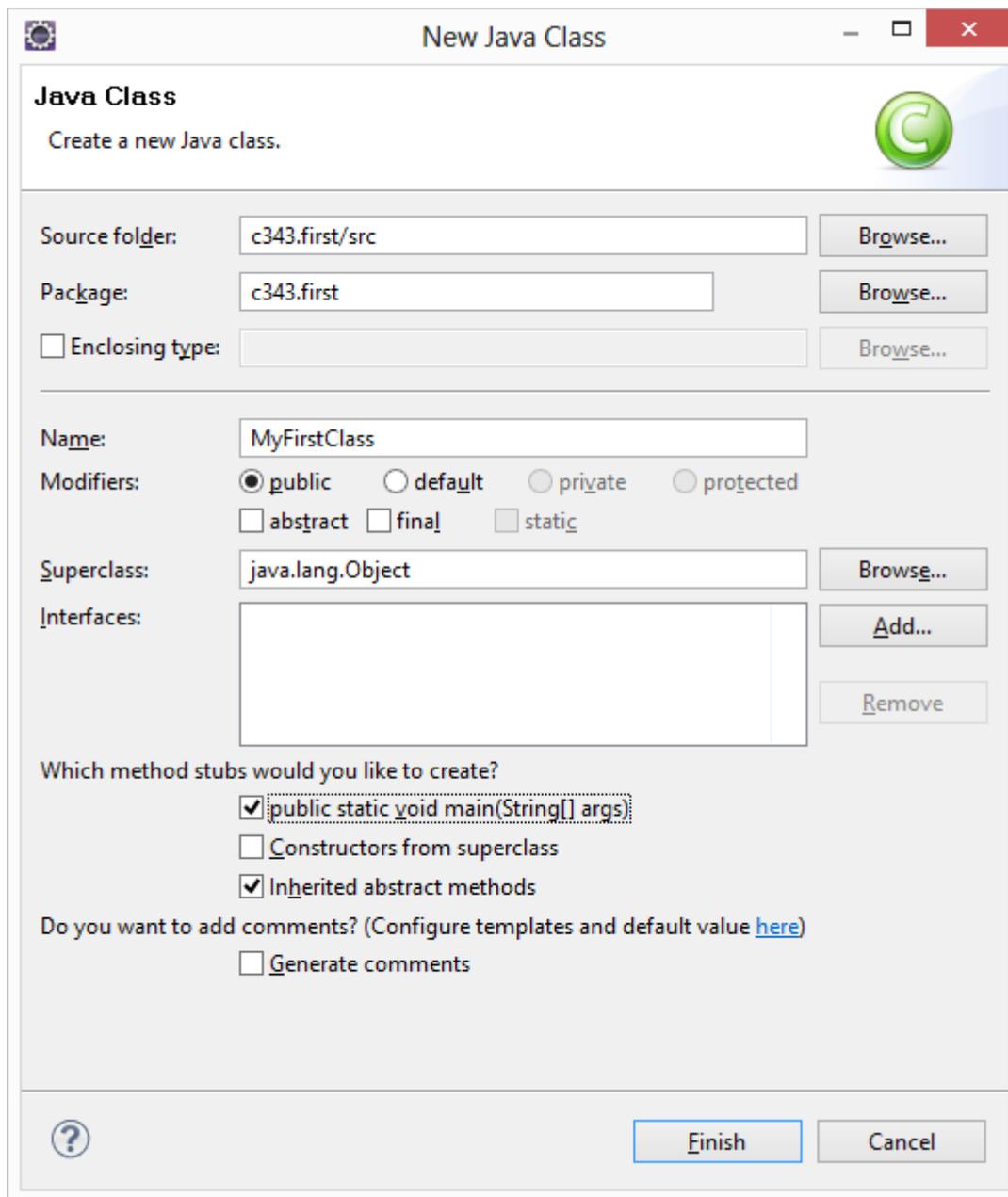
In the following step you create a new package. A good convention for the project and package name is to use the same name for the top level package and the project. For example, if you name your project `com.example.javaproject` you should also use `com.example.javaproject` as the top-level package name.

To create the `c343.first` package, select the `src` folder, 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

Right-click on your package and select *New* → *Class*. Enter `MyFirstClass` as the class name and select the *public static void main (String[] args)* checkbox. Press the *Finish* button.



This creates a new file and opens the Java *editor*. Change the class based on the following listing.

```
package c343.first;

public class MyFirstClass {

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

You could also directly create new packages via this dialog. If you enter a new package in this dialog, it is created automatically.

Run your project in Eclipse

Now run your code. Either right-click on your Java class in the *Package Explorer* or right-click in the 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.