Creating a Java project in Eclipse

When you start the Eclipse application, you see a window that looks like this. You may not see the Package Explorer pane initially; it may look like this. You can make that pane visible using a menu item from the top horizontal menu bar. This is where it is on the mac; on other machines it's in a slightly different place but easily seen. We use menu item Window -> Show View -> Package Explorer.

Now, we have:

1. The Package Explorer pane will contain Java projects.
2. The pane in the middle will later contain Java files that you are editing.
3. Some other parts, which we will look at later.

Creating a Java project

We show you how to create a Java project. First, use menu item File -> new -> Java Project.

In the window that opens, type a project name. Use the default location. Eclipse will store *all* projects in the default location unless you tell it differently. Note where it is! Use execution environment JavaSE-1.8. We suggest using separate folders for sources and class files —more on this later. Click *Finish*.

You see the project in the Package Explorer Pane. Click the horizontal arrow that is before the project name to see that the project contains a source (src) directory and a library.

Adding a Java class to the project

Let's add a Java class to the project. With the project selected, use menu item File -> New -> Class —or hover your mouse over the white c in the green circle until "New Java Class" appears and then click the mouse. The window opens for information about a new Java class.

The project will be put in directory firstDemo/src. Make the *Package* field empty. This is important! Give the class a name: Greetings. Click the box to create method main. Then click *Finish*.

The class has been created and placed in the so-called "default package", and the middle pane contains the generated class. The Outline pane contains a list of components of class Greetings. If there are several of them, clicking one will select that component in the editing pane. This is an easy way to get to a particular line in the editing pane.

Make method main do something

Let's put a comment in to say what method main will do. Now, we can write a statement to do what the comment says. The statement that will print a String x to the console pane is

```
System.out.println(x);
```

We replace x by the string literal that we want to print. Before starting this video, I placed in my clipboard a greeting in the 5 most popular languages: Chinese, Spanish, English, Hindi, and Arabic. So let me paste that line in here, surrounded by double quotes. After the video, we'll show you more about the greetings.

By the way, I change the sizes of the various panes by placing the mouse between panes, pressing the mouse button, and dragging.

Executing (running) the Java application

Because class Greeting has a static procedure main, it is an application and can be executed, or run. Running this class consists of calling method main. We can run it simply by using menu item Run -> Run or by clicking the white arrow in the green circle. Let's click the arrow.

We have not saved the file yet, so a window pops up, asking us whether it should be saved. Let's check the box that says to *always* save resources before running a program. Now, we will never have to see this window again.

Now, click the OK button. That window disappears, the program is run, and the results of execution are shown in the Console pane.

We discuss main's parameter later

Method main has a parameter, of type String[]. You do not yet know about arrays in Java, and you don't know how to give an argument to method main. We'll talk about this much later.