

Creación de sus primeros proyectos Java en Eclipse

Introducción

Acerca de los paquetes Java

- Los paquetes Java se utilizan para identificar fácilmente clases donde no haya conflictos de nombres, así como para mantener juntos los archivos .java relacionados
- La API de Java contiene muchos paquetes
- La API de Java es una biblioteca de clases ya escritas que se pueden utilizar gratuitamente y que están disponibles en Java Runtime Environment
- También podrá crear sus propias clases que se conservarán mejor en el paquete que cree
- Si no crea un paquete para estas, NetBeans, Eclipse u otros IDE utilizarán el paquete por defecto
- **Para los ejemplos de este curso, a menudo utilizará el paquete por defecto cuando desee probar los ejemplos de codificación con las instrucciones de la parte 1**
- Para la **parte 2**, creará un archivo .java dentro de un paquete denominado **mysecondproject**
- Cuando crea un archivo .java en un paquete llamado **mysecondproject**, necesita que la declaración del paquete sea su primera línea de código:
 - `package mysecondproject;`
- Si prefiere utilizar el paquete por defecto, no se utiliza una declaración de paquete

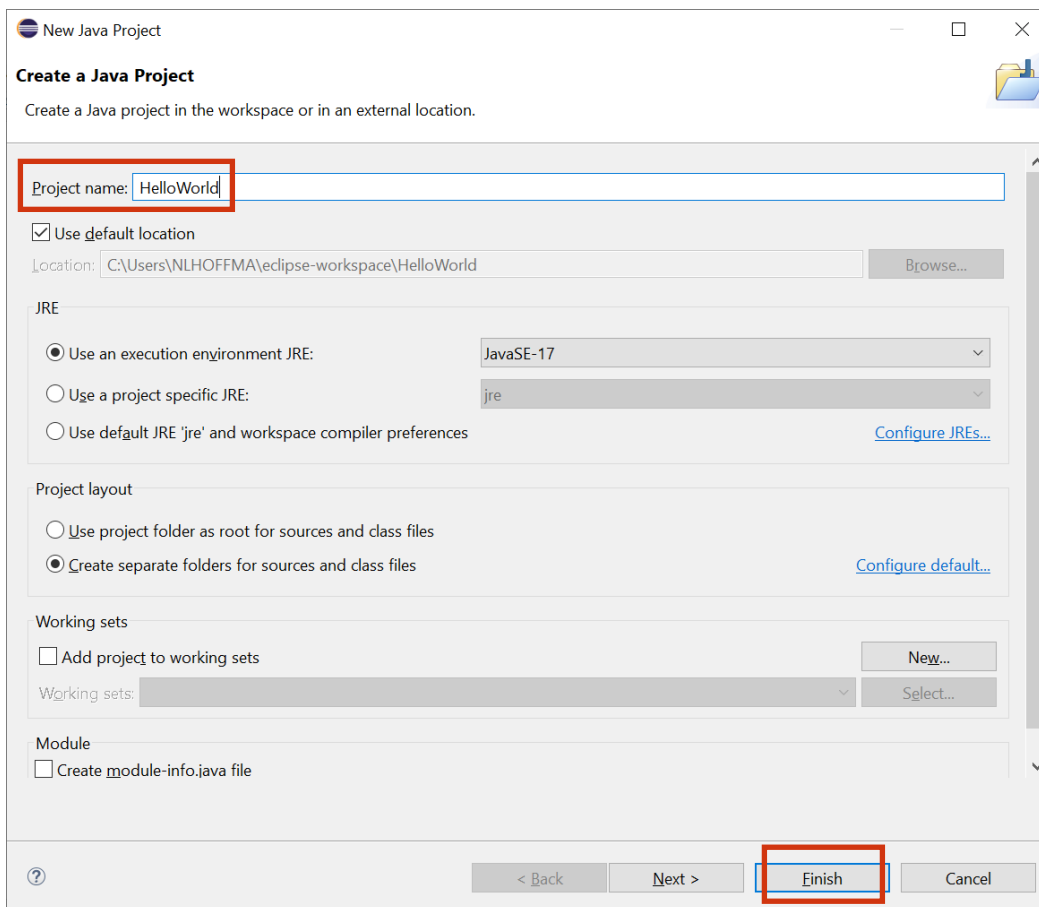
Parte 1: Crear un proyecto Java y agregar un archivo .java existente

Visión general

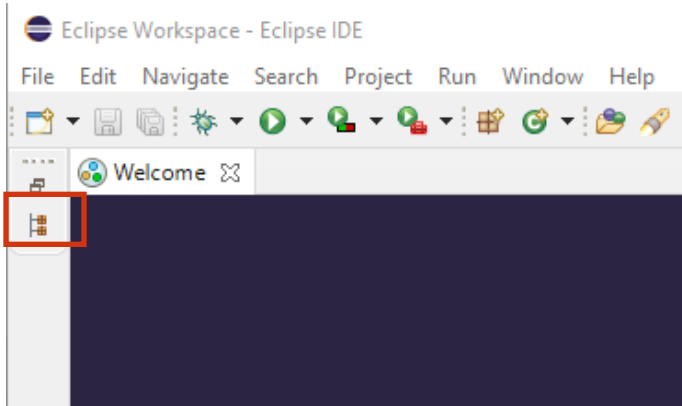
En esta sección, descargará el material de referencia HelloWorld.zip, creará un proyecto nuevo en Eclipse y agregará un archivo Java existente al proyecto.

Tareas

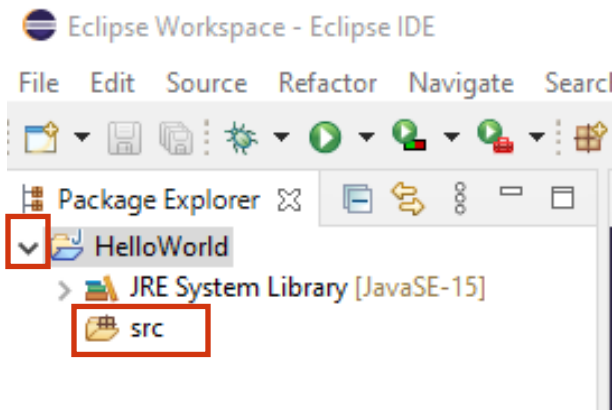
1. Descargue el archivo HelloWorld.zip de los materiales de referencia para esta práctica
2. Extraiga el archivo zip HelloWorld.zip y anote la ubicación
3. Inicie Eclipse
4. Vaya a **File > New > Java Project**
 - a. Agregue el nombre del proyecto: HelloWorld
 - b. Para este curso, se recomienda que asigne a los proyectos el mismo nombre que al archivo .java
5. Haga clic en **Finish**



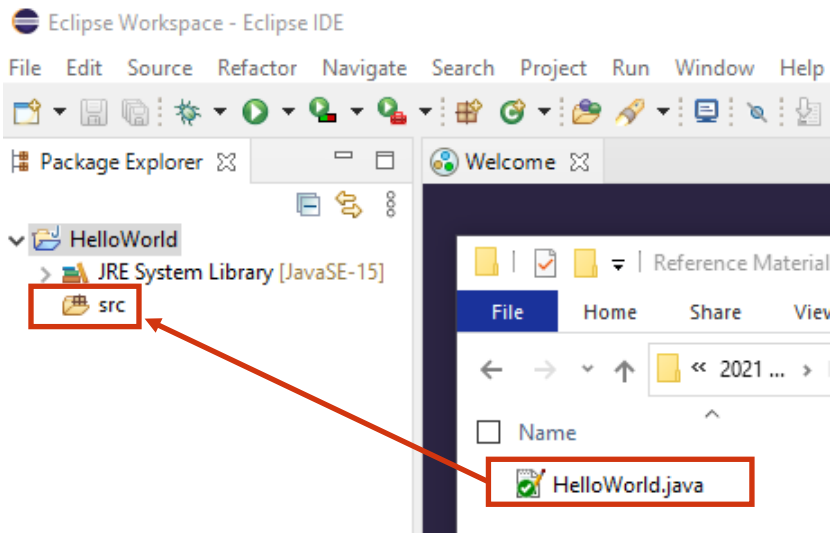
6. Haga clic en el icono **Restore** para ver la ventana Package Explorer (si no está abierta)



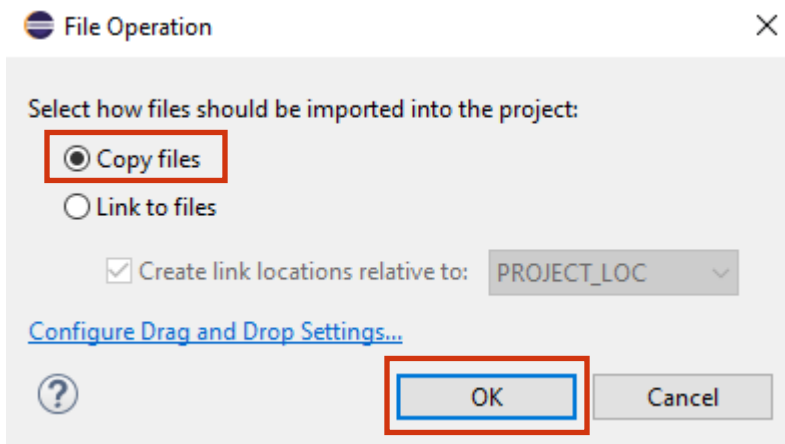
7. En la ventana Package Explorer, **haga clic en la flecha** para ampliar el proyecto y ver el paquete src



8. **Desplácese a la carpeta** en la que haya extraído el archivo HelloWorld.java en el paso 2
9. **Arrastre** el archivo HelloWorld.java desde la carpeta del dispositivo y **suéltelo** en carpeta **src**

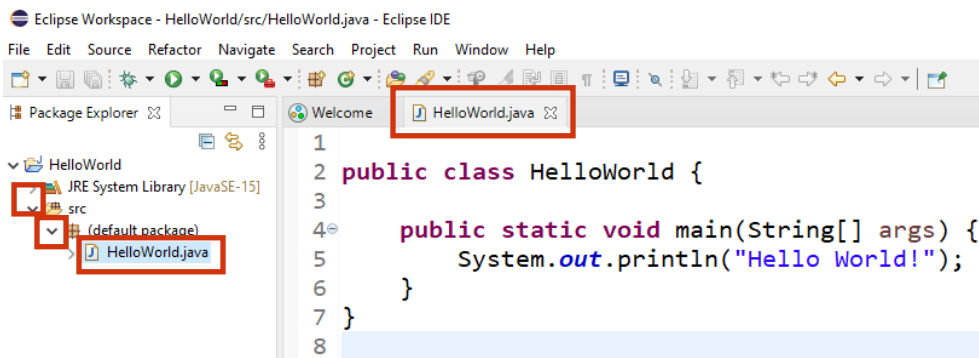


10. Cuando se le solicite, **seleccione Copy files y haga clic en OK**

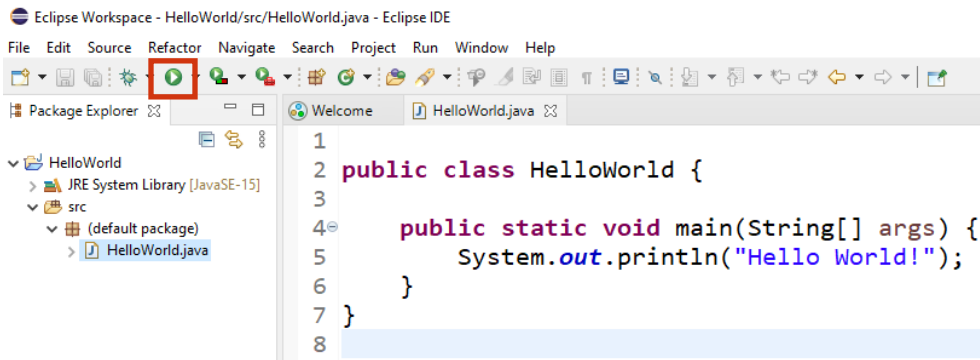


11. **Haga clic en la flecha** situada junto a la carpeta src y, a continuación, **haga clic en la flecha** situada junto a (default package) para ver el archivo HelloWorld.java en el paquete por defecto

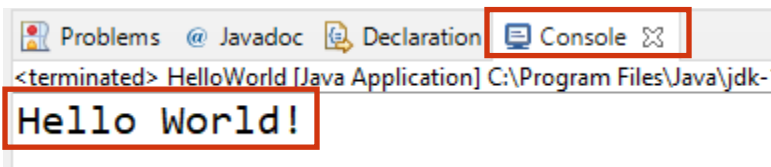
12. **Haga doble clic** en el archivo .java para que se abra en un nuevo separador en la ventana del editor de código



13. Haga clic en el botón Run para realizar la prueba



14. En la ventana de la consola debajo del editor de código, debería ver el mensaje "Hello World"



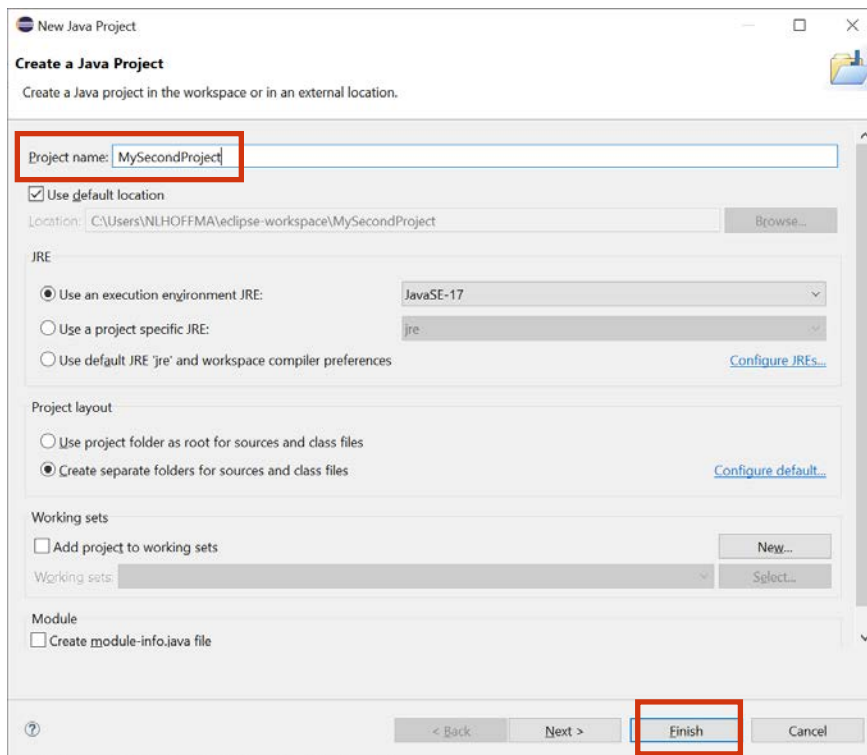
Parte 2: Crear un paquete y un proyecto en Eclipse con una clase main

Visión general

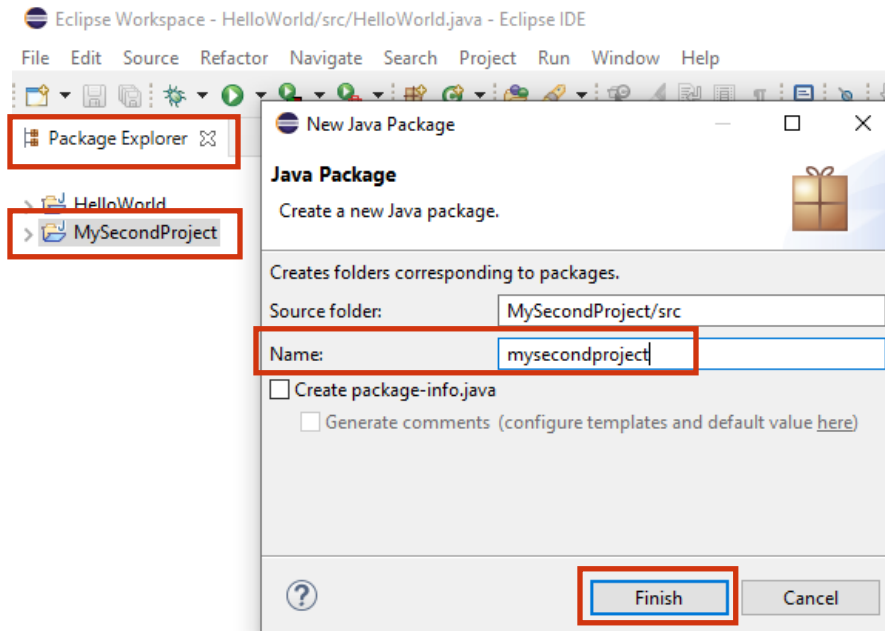
En esta sección, creará y ejecutará un proyecto java de Eclipse dentro de un paquete con una clase Java main

Tareas

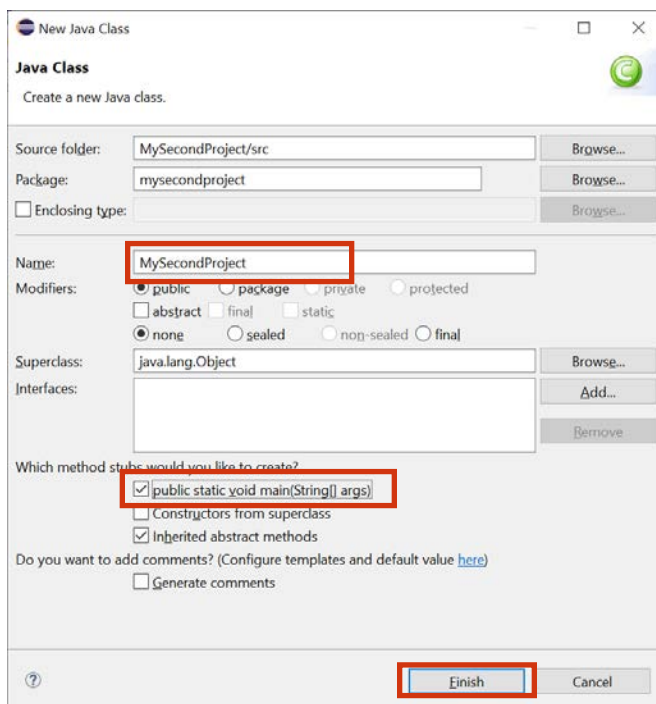
1. Inicie Eclipse si no está abierto
2. Vaya a **File > New > Java Project** y seleccione lo siguiente:
 - a. Agregue el nombre del proyecto - MySecondProject
3. Haga clic en **Finish**



4. En Package Explorer, **haga clic con el botón derecho** en el proyecto **MySecondProject** > **haga clic en New > Package**
5. Escriba el nombre **mysecondproject** y, a continuación, haga clic en **Finish**



6. En Package Explorer, **haga clic con el botón derecho** en el paquete **mysecondproject** > **haga clic en New > Class**
7. Escriba el nombre **MySecondProject** y, a continuación, active la casilla para crear **public static void main**. Haga clic en **Finish**



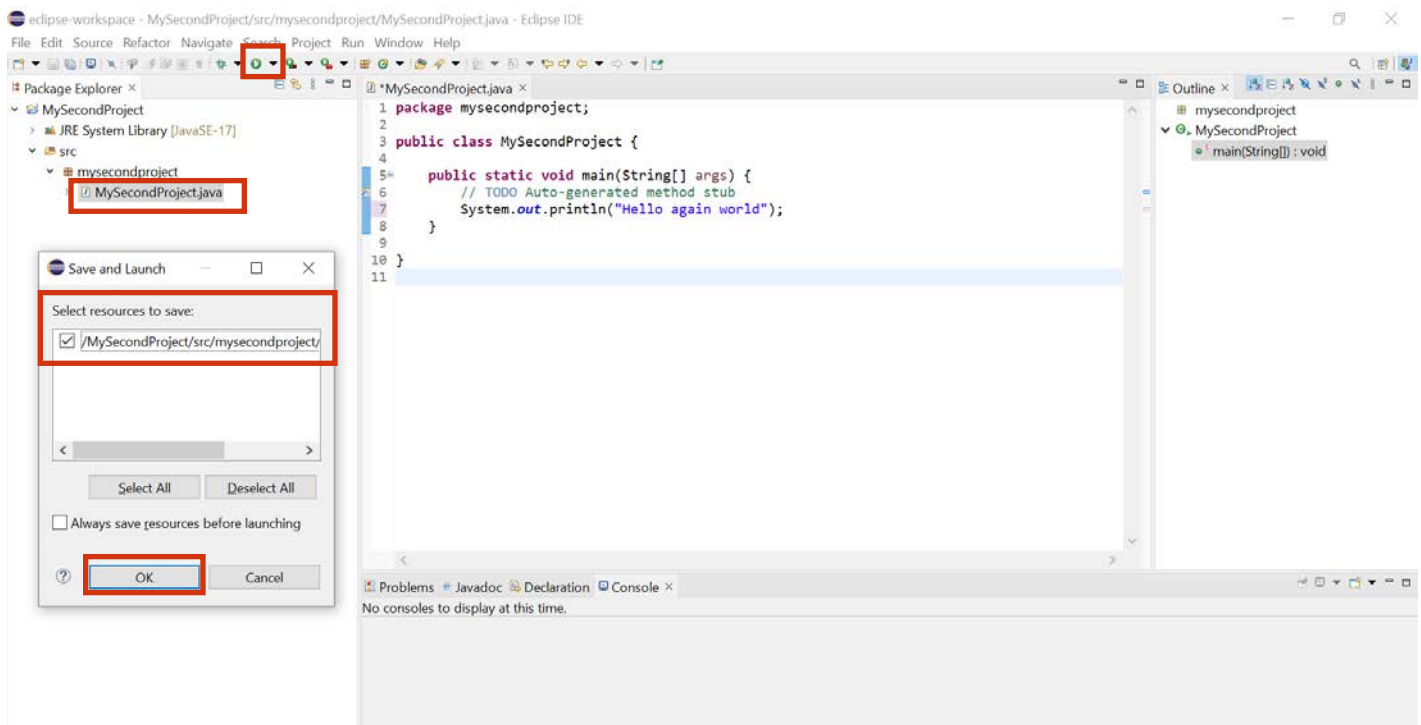
8. La clase MySecondProject recién creada se abrirá en un nuevo separador del editor de código
9. En el editor de código, busque el método main de la clase MySecondProject e **introduzca la línea de código** como se muestra a continuación:

System.out.println("Hello again world");

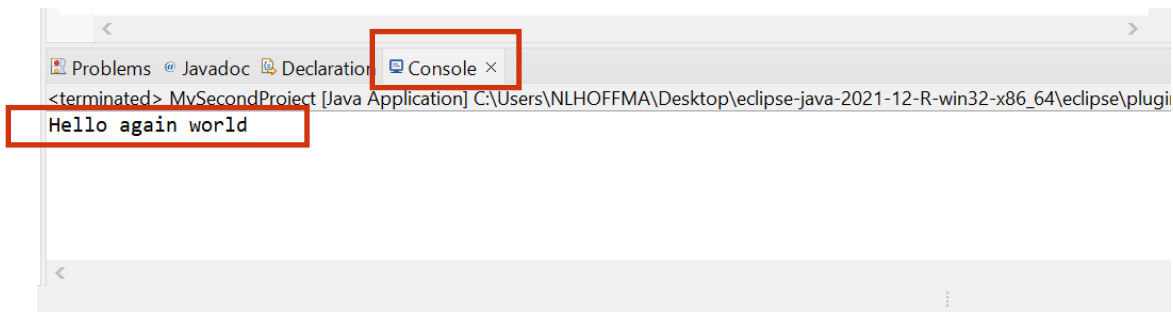


```
1 package mysecondproject;
2
3 public class MySecondProject {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello again world");
8     }
9
10 }
11
```

10. Asegúrese de que la clase MySecondProject.java está resaltada en Package Explorer y, a continuación, **haga clic en el botón Run** para probar (o haga clic en la pequeña flecha hacia abajo a la derecha del botón Run y seleccione el archivo Java que desea ejecutar) - Si se le solicita, **haga clic en OK** para guardar los recursos

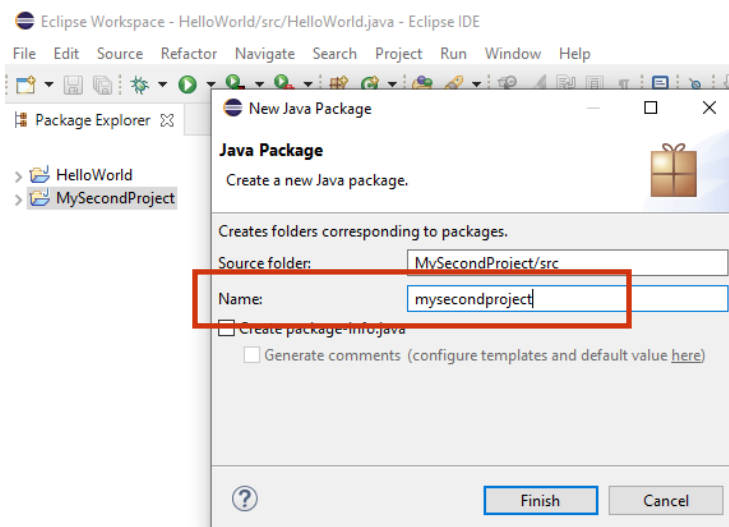


11. Debería ver el mensaje “Hello again world” en la salida de la consola en la parte inferior del IDE



Observe una diferencia importante entre el primer proyecto y el segundo:

En la tarea 5 de MySecondProject, agregó un paquete explícitamente y le asignó el nombre mysecondpackage.

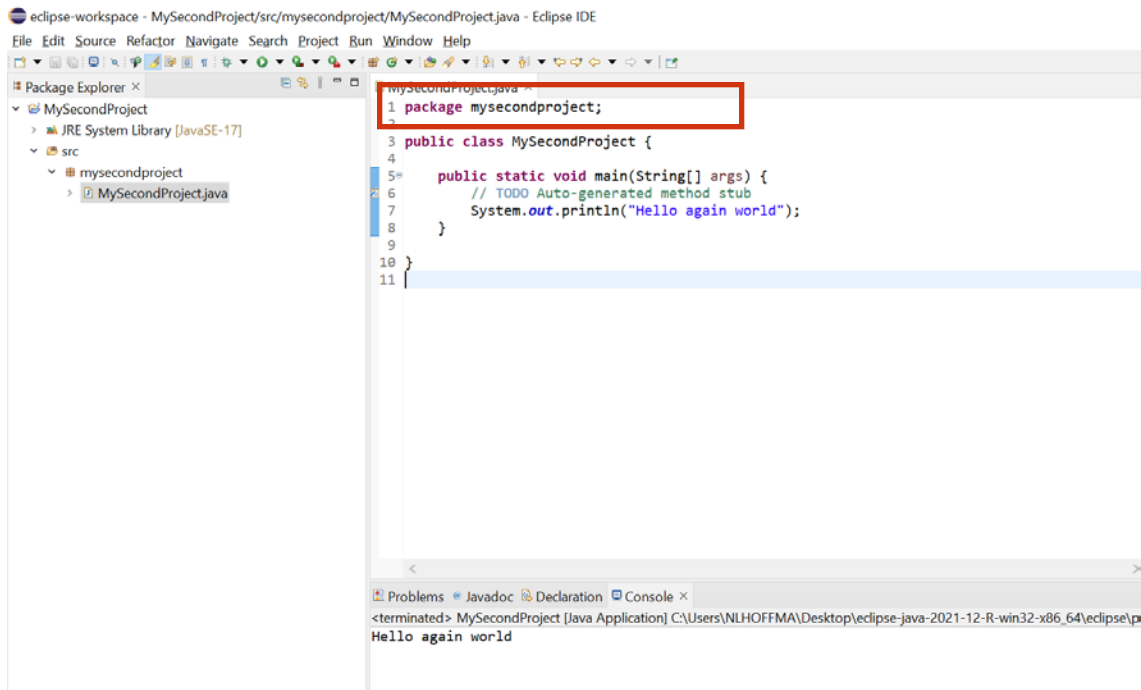


En la parte 1, no creó una clase main, simplemente arrastró un archivo .java existente a la carpeta src. Eclipse creó implícitamente el paquete y le asignó el nombre (default package)

Si un paquete tiene un nombre que no sea (default package), es importante tener en cuenta que se debe incluir una declaración de paquete como primera línea de código en el archivo Java

Si crea explícitamente una clase en un paquete existente, Eclipse agregará automáticamente la declaración del paquete

En el editor de código del segundo proyecto, desplácese hasta el principio del código y verá la declaración del paquete



Observe que, en su primer proyecto, como el paquete era (default package), si se desplaza a la parte superior del archivo java en el editor de código, verá que no hay ninguna declaración de paquete

Si agrega un archivo Java existente a un paquete al que ya se le ha asignado un nombre, deberá agregar la declaración del paquete manualmente como la primera línea de código del archivo Java con el formato:

```
package nombrepaquete;
```