

Creación de sus primeros proyectos Java en NetBeans

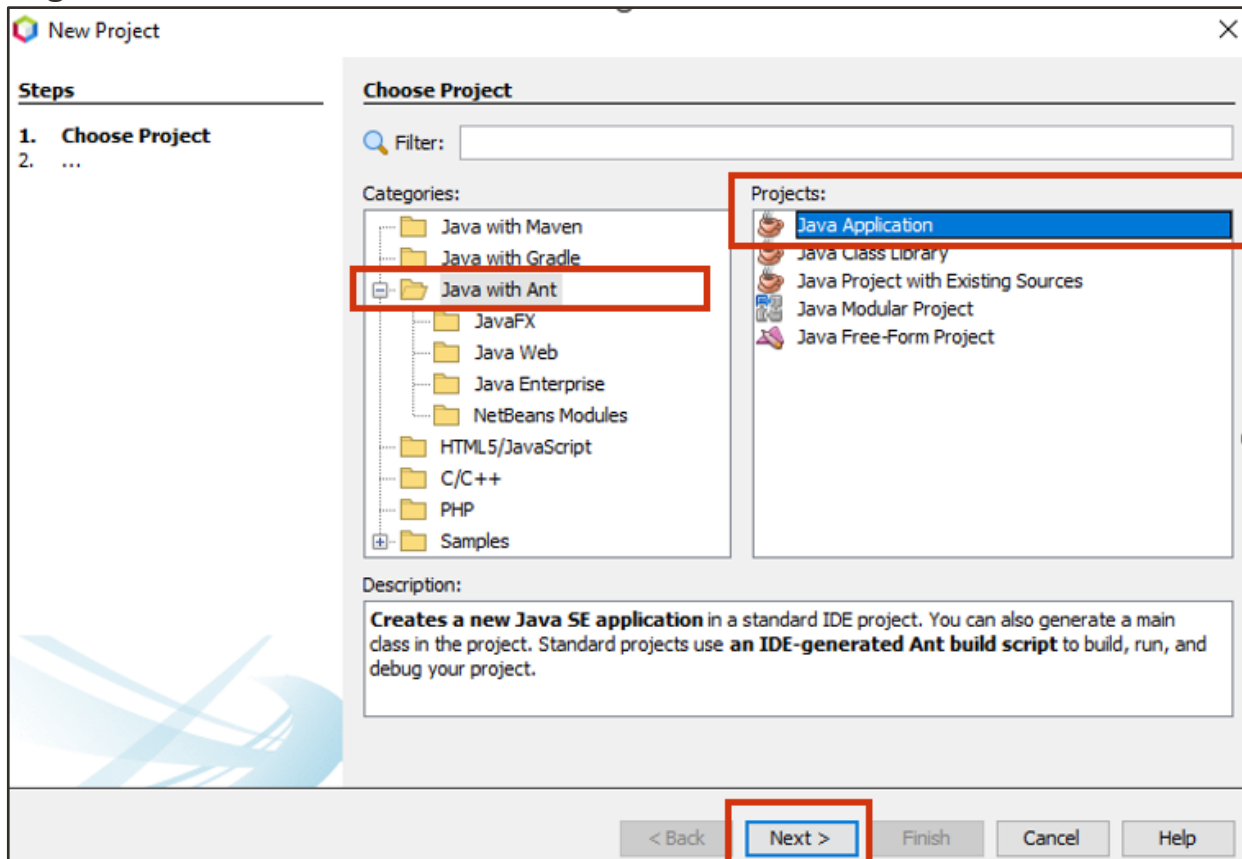
Parte 1: Creación de un proyecto Java y adición de un archivo .java existente

Descripción general

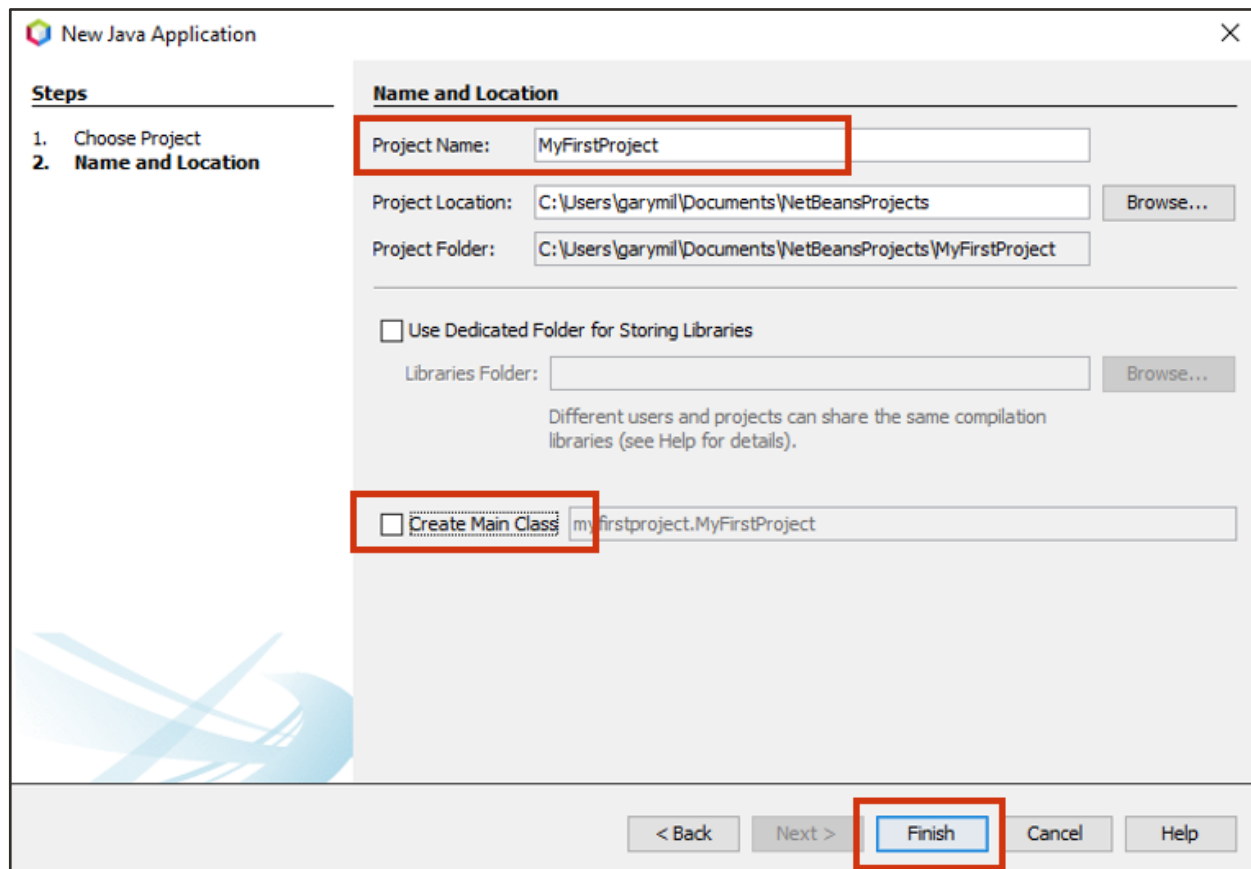
En esta sección, descargará el archivo **HelloWorld.zip**, creará un nuevo proyecto en NetBeans y agregará un archivo **.java** existente al proyecto.

Tareas

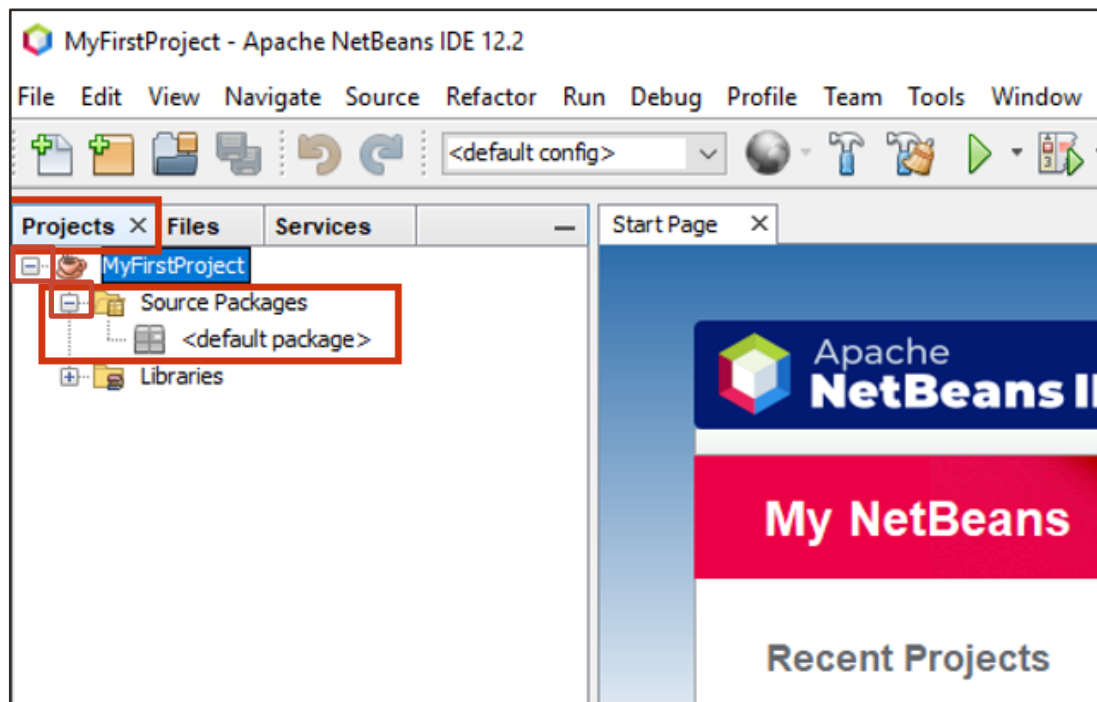
1. Descargue el archivo zip **HelloWorld.zip** para esta práctica. Extraiga el archivo zip y preste atención a la ubicación.
2. Inicie **NetBeans**.
3. Vaya a **File > New Project** y seleccione lo siguiente:
 - a. Categories: **Java with Ant**
 - b. Projects: **Java Application**
4. Haga clic en **Next >**.



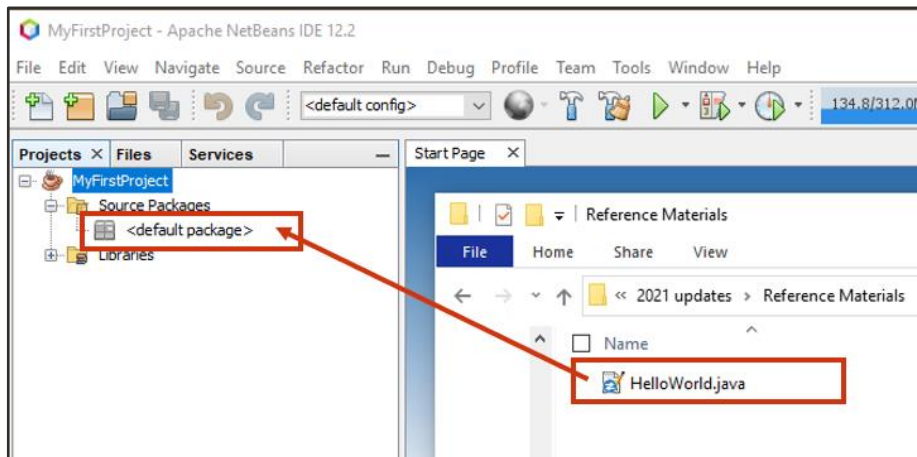
5. Asigne al proyecto el nombre **MyFirstProject**. **Desmarque** la casilla **Create Main Class** y haga clic en **Finish**.



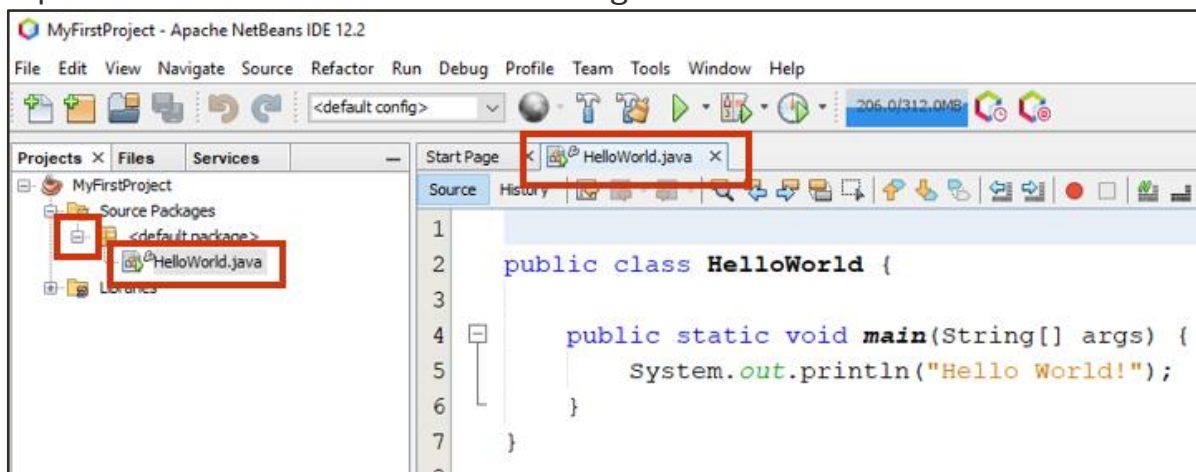
6. En la ventana **Projects**, haga clic en el **signo más** para ampliar el proyecto y de nuevo en el **signo más** para ampliar **Source Packages** y mostrar **<default package>**.



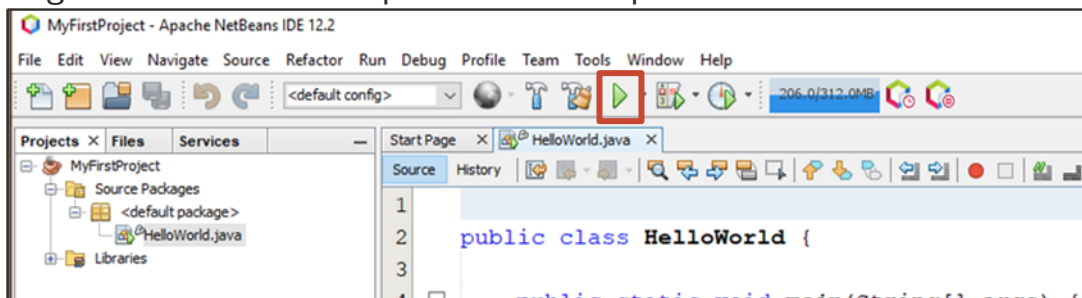
7. Vaya a la carpeta donde ha extraído el archivo **HelloWorld.zip** en el paso 1 y seleccione el archivo **HelloWorld.java**.
8. Arrastre el archivo **HelloWorld.java** desde la carpeta de su dispositivo y suéltelo en **<default package>**.



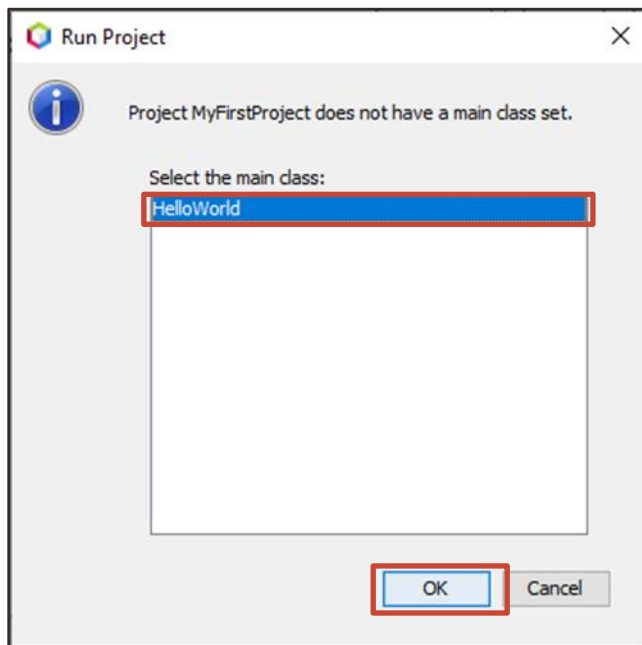
9. Haga clic en el **signo más** junto a **<default package>** y verá el archivo **HelloWorld.java** dentro del paquete. Haga doble clic en el archivo **HelloWorld.java** y se abrirá en un nuevo separador en la ventana del editor de código.



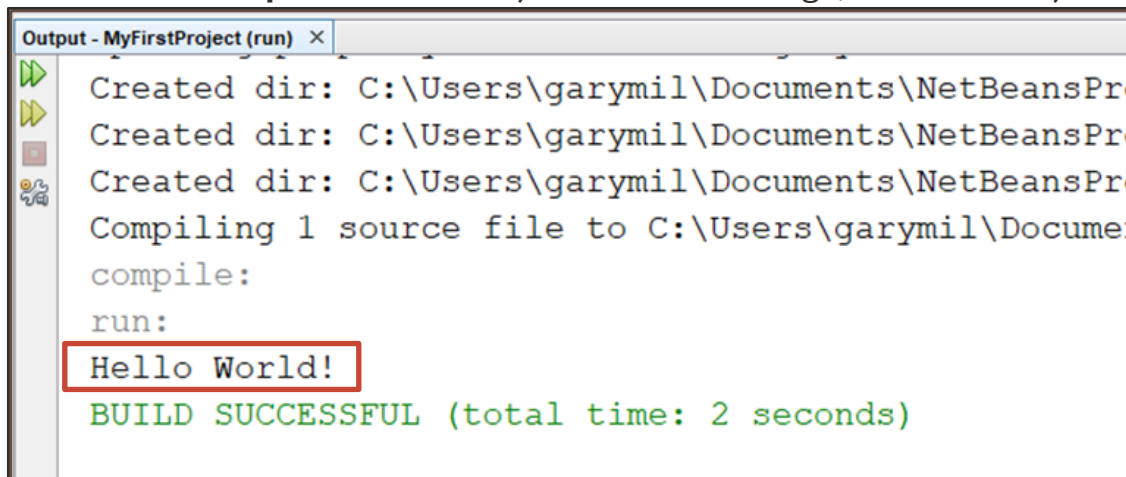
10. Haga clic en el botón **Run** para realizar una prueba.



11. Si se le solicita **Select the main class**, seleccione la opción por defecto **HelloWorld** y haga clic en **OK**.



12. En la ventana **Output** ubicada debajo del editor de código, verá el mensaje "**Hello World!**".



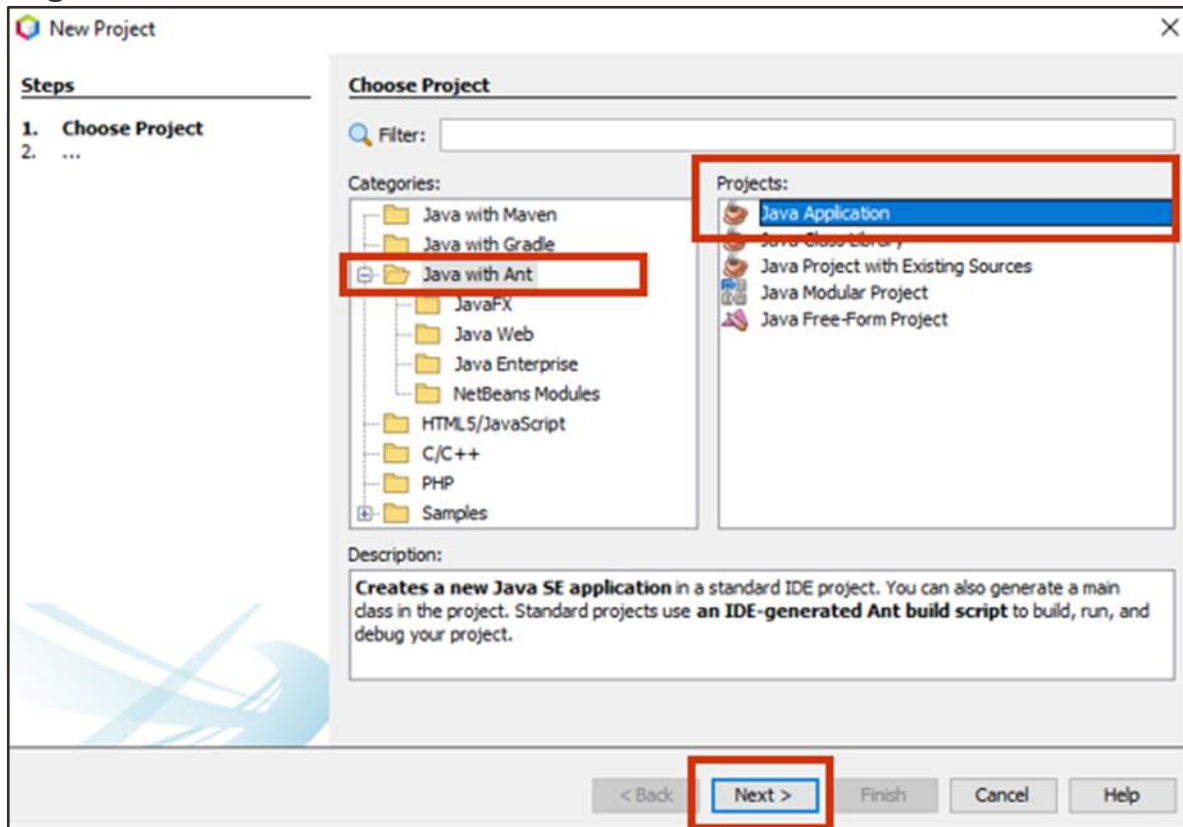
Parte 2: Creación de un proyecto en NetBeans con una clase main

Descripción general

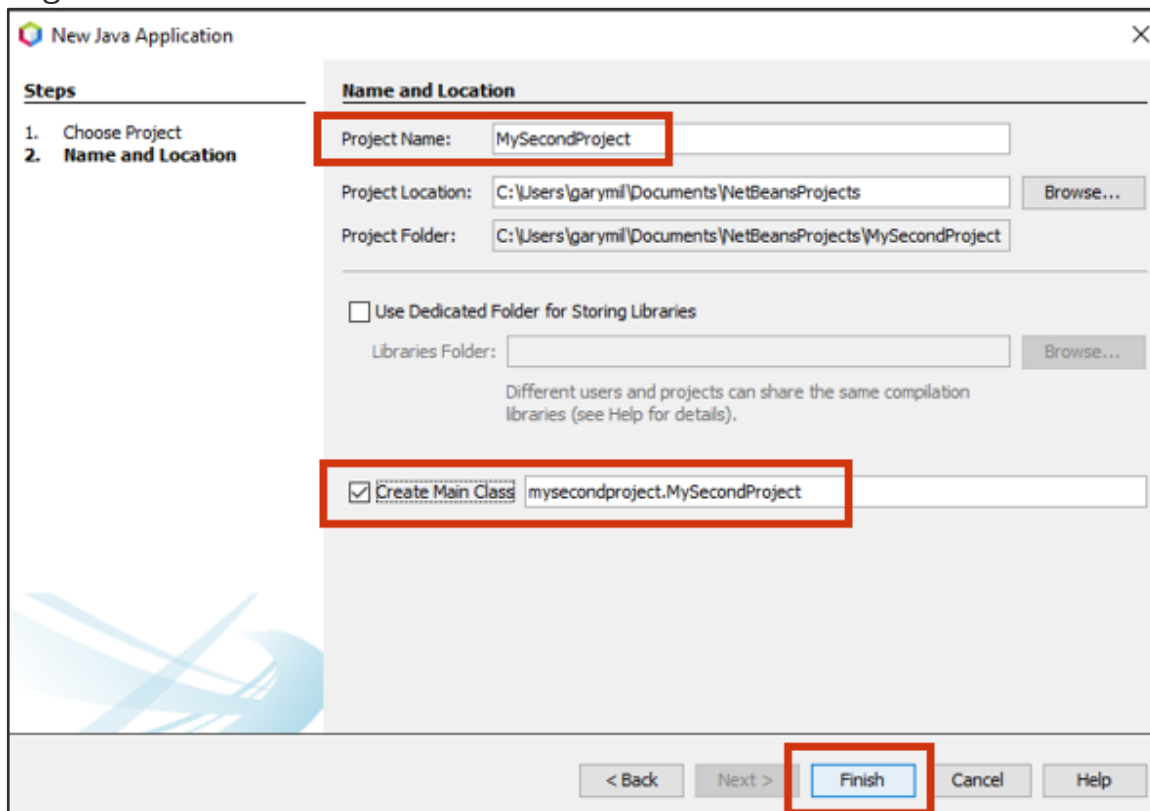
En esta sección, creará y ejecutará un proyecto de Java en NetBeans con una clase Java del tipo main.

Tareas

1. Inicie **NetBeans** si no lo tiene abierto.
2. Vaya a **File > New Project** y seleccione lo siguiente:
 - a. Categories: **Java with Ant**
 - b. Projects: **Java Application**
3. Haga clic en **Next >**.

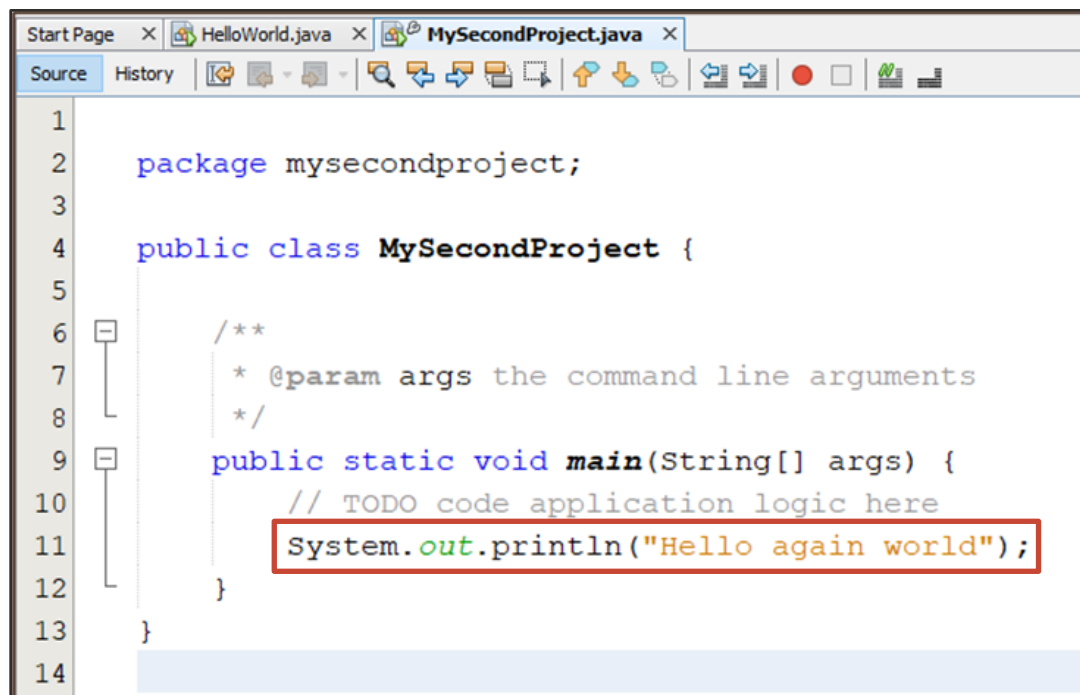


- Asigne al proyecto el nombre **MySecondProject**. Marque la casilla **Create Main Class** y haga clic en **Finish**.

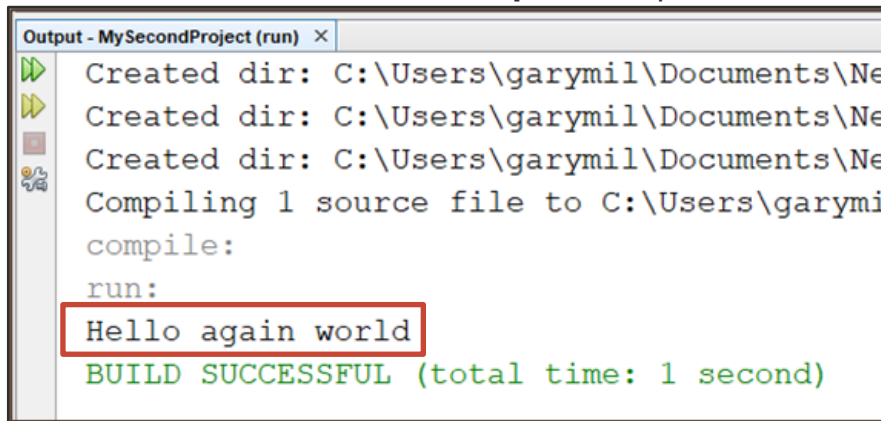


- En el editor de código, localice el método main de la clase **MySecondProject** e introduzca la línea de código siguiente:

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

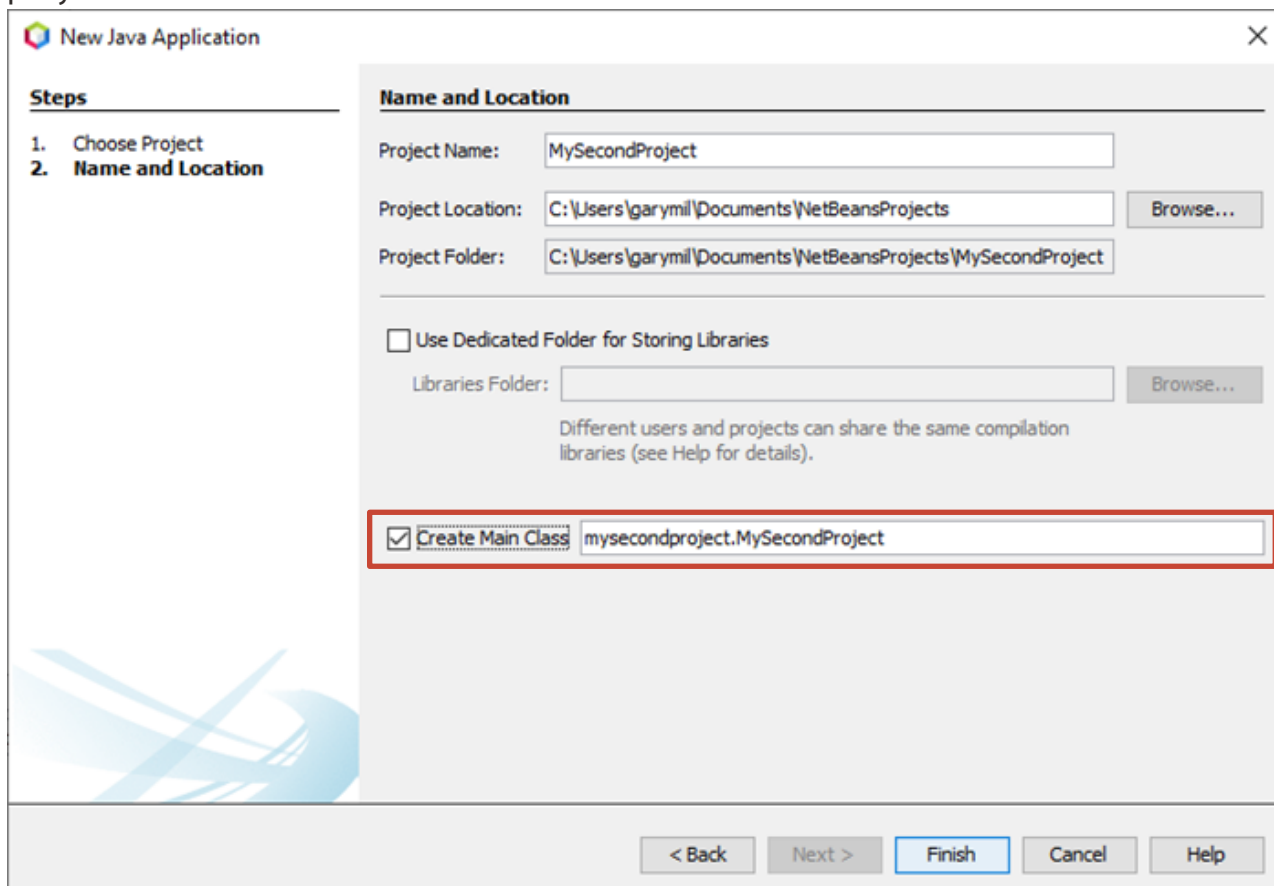


6. Haga clic en el botón **Run** para realizar una prueba. Debería ver el mensaje "**Hello again world**" mostrado en la consola **Output**, en la parte inferior del IDE.



```
Output - MySecondProject (run) x
Created dir: C:\Users\garymil\Documents\Ne
Created dir: C:\Users\garymil\Documents\Ne
Created dir: C:\Users\garymil\Documents\Ne
Compiling 1 source file to C:\Users\garymi
compile:
run:
Hello again world
BUILD SUCCESSFUL (total time: 1 second)
```

Observe una diferencia importante entre el primer proyecto y el segundo. En **MySecondProject**, cuando se marca la casilla **Create Main Class** en la Tarea 4, también se crea un paquete con nombre. NetBeans agrega un nombre sugerido de clase y de paquete que se basa en el nombre del proyecto.



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: MySecondProject

Project Location: C:\Users\garymil\Documents\NetBeansProjects Browse...

Project Folder: C:\Users\garymil\Documents\NetBeansProjects\MySecondProject

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

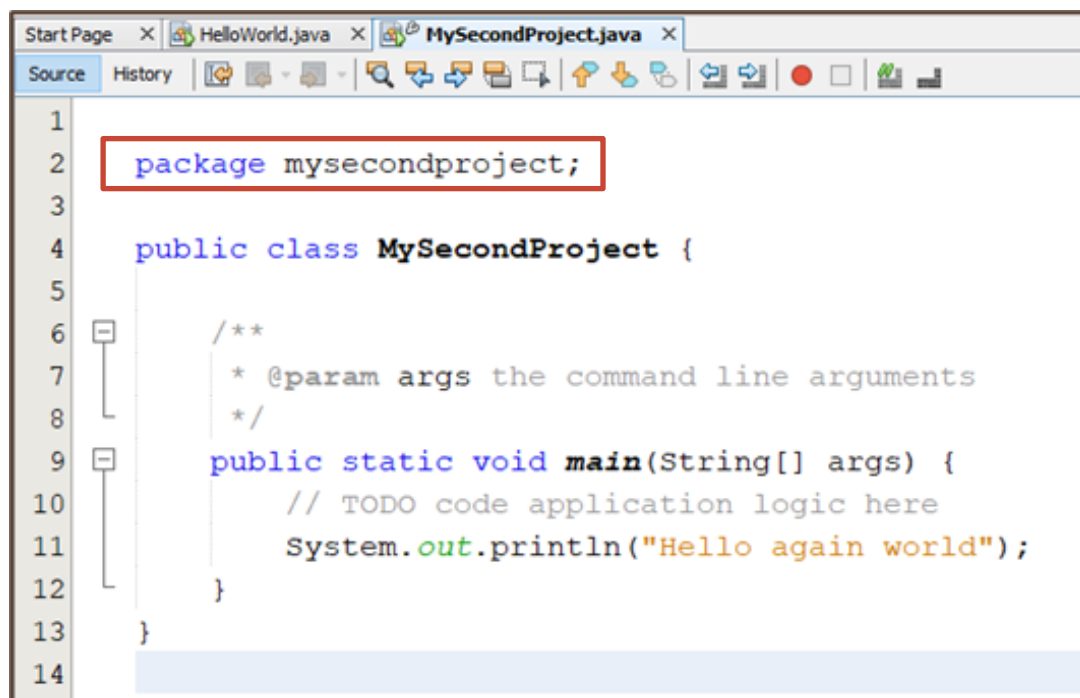
Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class mysecondproject.MySecondProject

< Back Next > Finish Cancel Help

El formato de nombre sugerido es **packagename.MainClassName**, en este caso, **mysecondproject.MySecondProject**. Puede editar los nombres en este punto si desea poner otro nombre distinto del formato de nombre sugerido.

En la Parte 1, como no creó una clase main, el paquete recibió el nombre **<default package>**. Si el paquete tiene un nombre distinto a **<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 una clase en un paquete existente, NetBeans agregará automáticamente la declaración de paquete por usted. En el editor de código del segundo proyecto, desplácese al principio del código y verá la declaración de paquete.



```
1
2 package mysecondproject;
3
4 public class MySecondProject {
5
6     /**
7      * @param args the command line arguments
8      */
9     public static void main(String[] args) {
10         // TODO code application logic here
11         System.out.println("Hello again world");
12     }
13 }
14
```

En el primer proyecto, como el paquete era **<default package>**, si se desplaza al principio 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 que ya tiene nombre, deberá agregar la declaración de paquete manualmente como la primera línea de código del archivo **.java** usando el siguiente formato:

```
package packagename;
```