

Using NetBeans™ to Compile and Run Java Programs

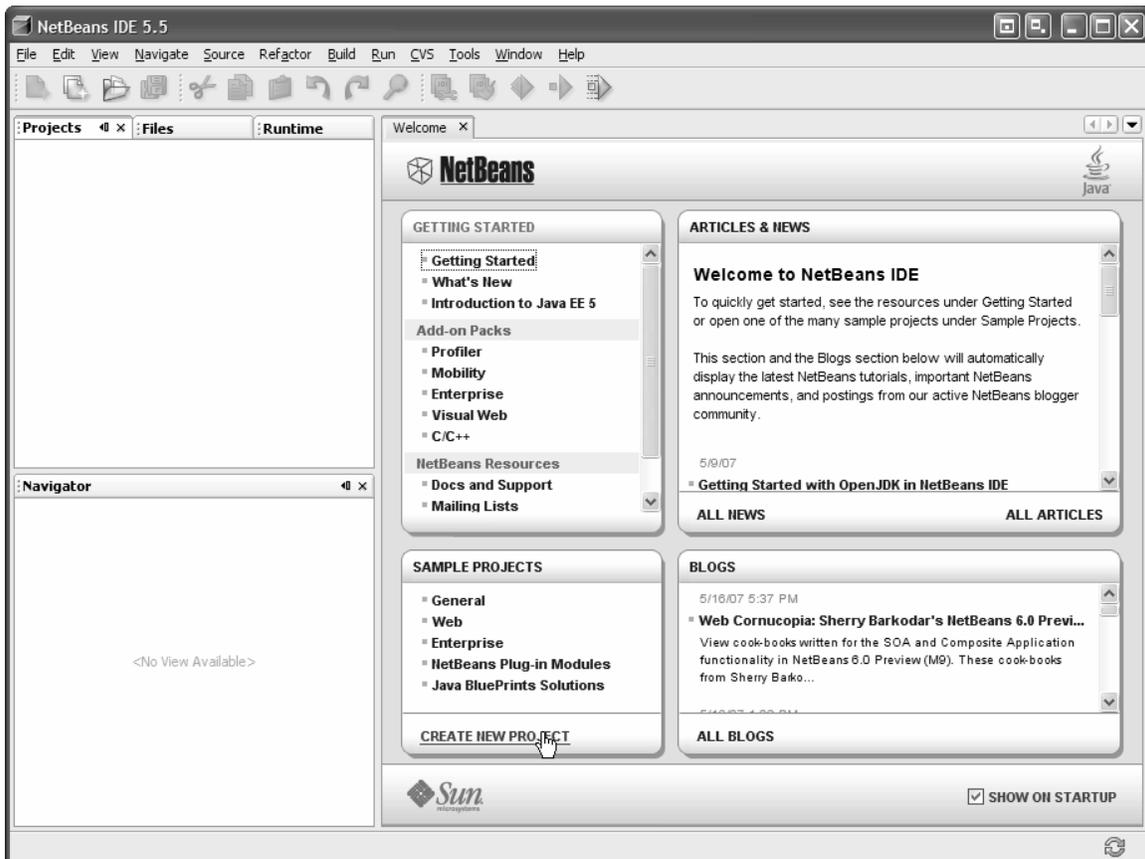
This document is based on NetBeans 4.1 and 5.5. (The document has two sections. The first section covers NetBeans 5.5; the second section covers NetBeans 4.1. But really, the instructions for NetBeans 4.1 and NetBeans 5.5 are almost identical.) For information on other versions of NetBeans, contact the author via email: [BeginProg2 at BurdBrain dot com](mailto:BeginProg2@BurdBrain.com).

NetBeans 5.5

1. Download NetBeans by visiting <http://www.netbeans.org>.
2. Install the downloaded NetBeans program on your computer.
3. Run NetBeans by clicking the NetBeans icon.

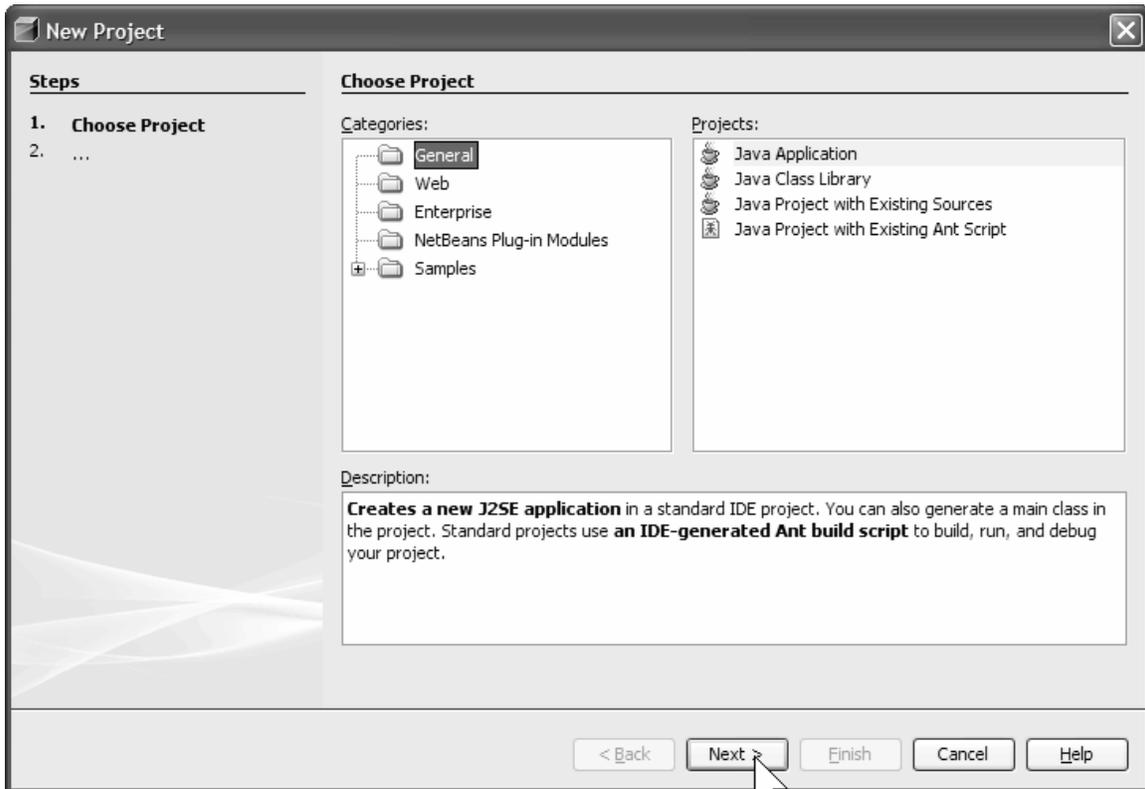


You see the NetBeans 5.5 opening screen.



4. Click Create New Project.

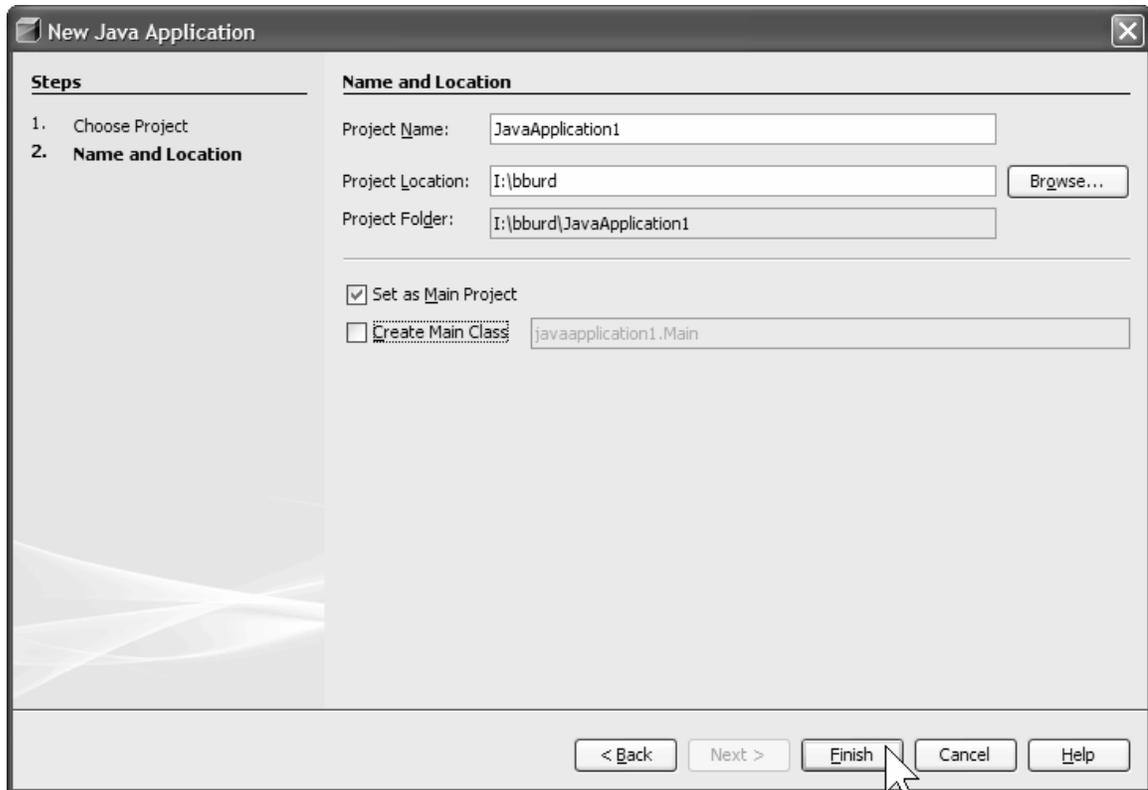
The New Project dialog appears.



5. In the Categories pane, select General, and in the Projects pane, select Java Application.

6. Click Next.

The New Java Application dialog appears.



7. Unless you're fussy, you can accept whatever defaults you see in the Project Name, Project Location, and Project Folder fields. Leave Set as Main Project checked. But in this example, you should uncheck Create Main Class.

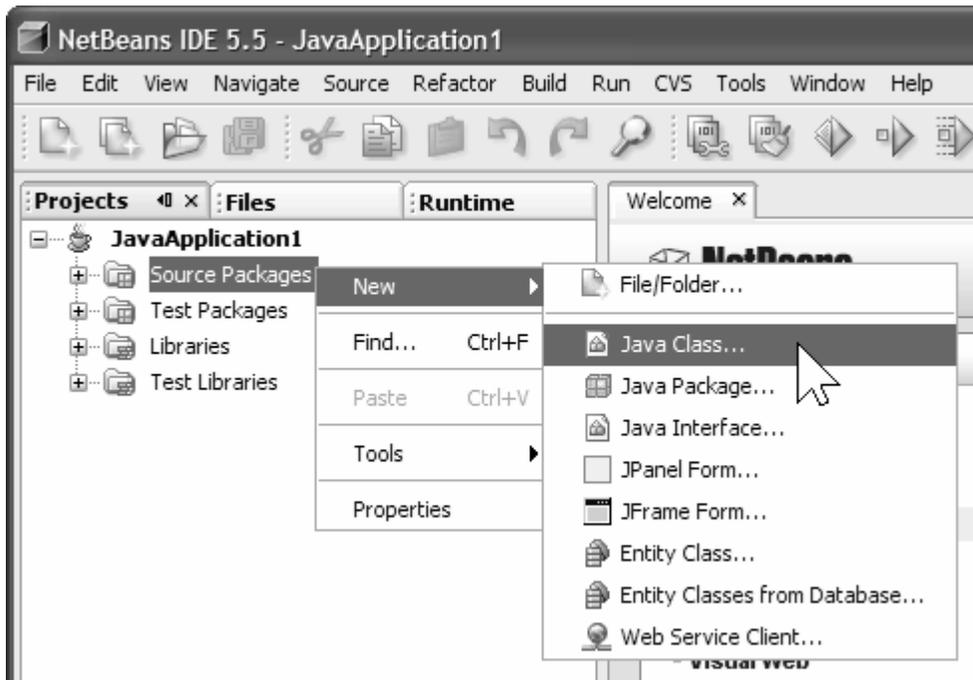
8. Click Finish.

A new project tree appears in the NetBeans Projects window.

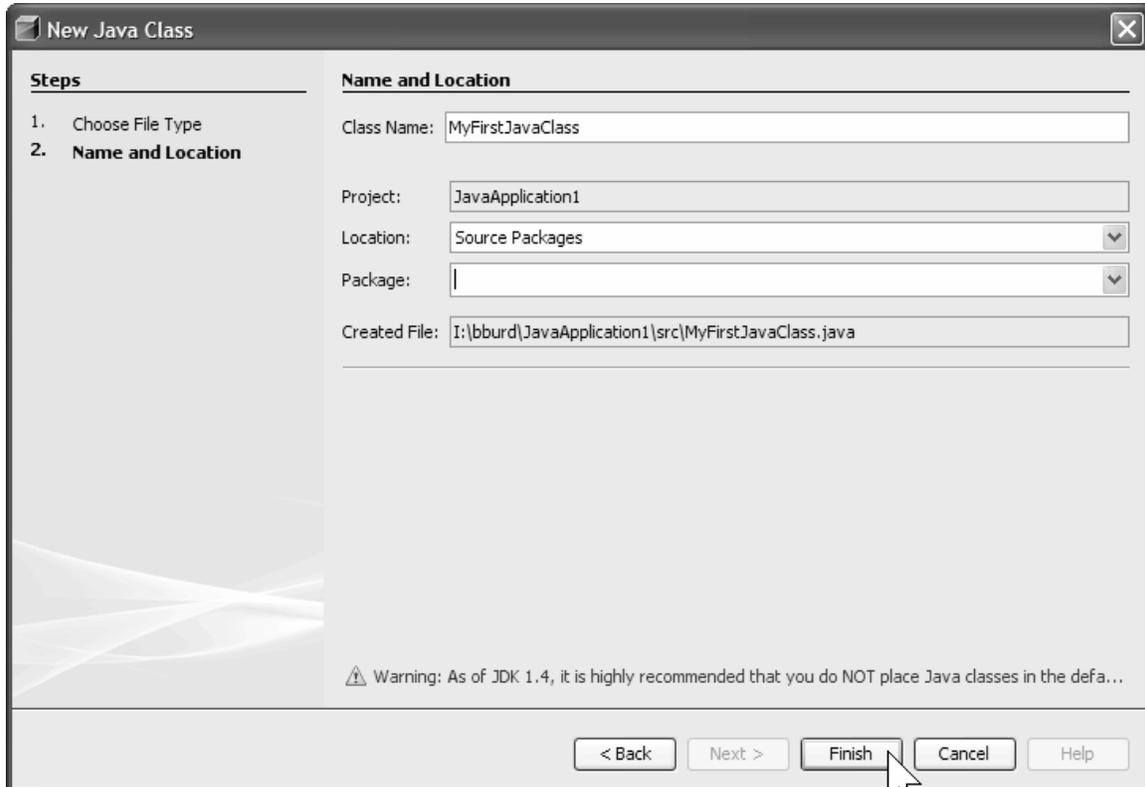


9. On the project tree, right-click the Source Package branch. In the resulting context menus, select New → Java Class.

On a Macintosh, substitute Ctrl-Click for right-click.



The New Java Class dialog appears.

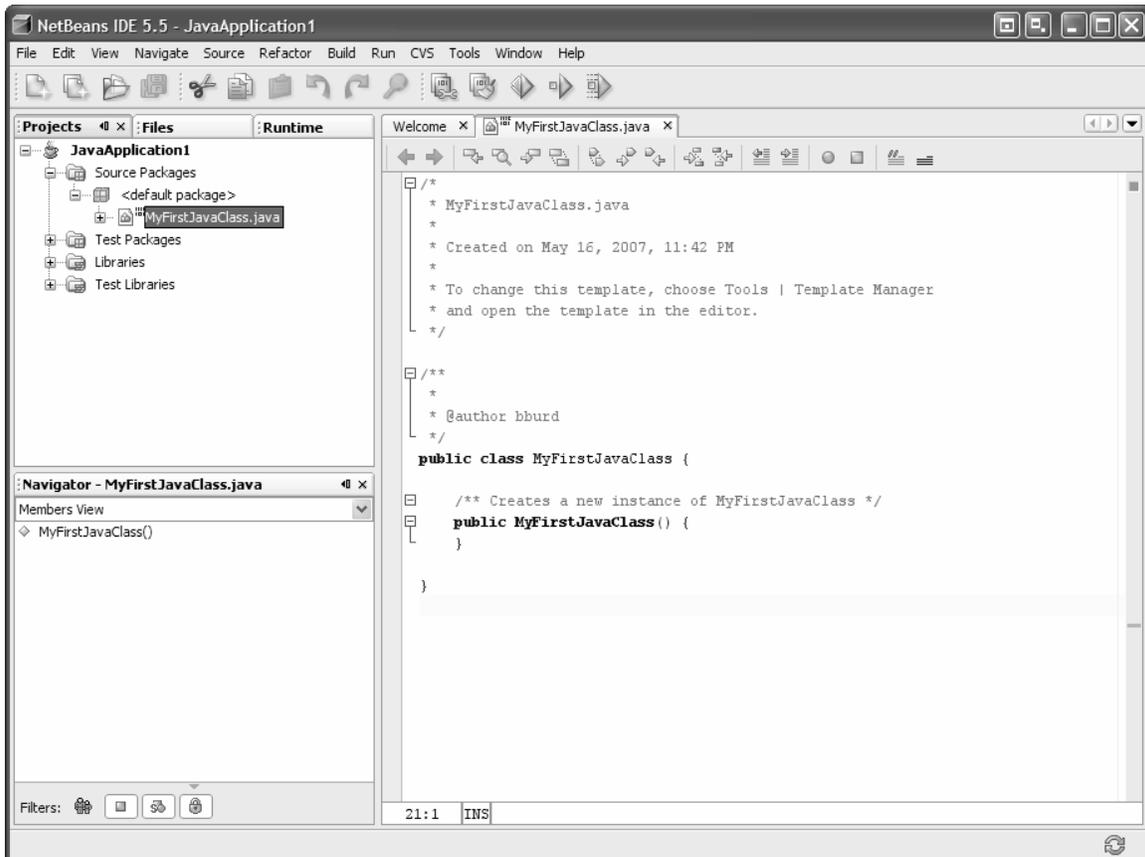


10. In the Class Name field, type the name of the class that you want to create.

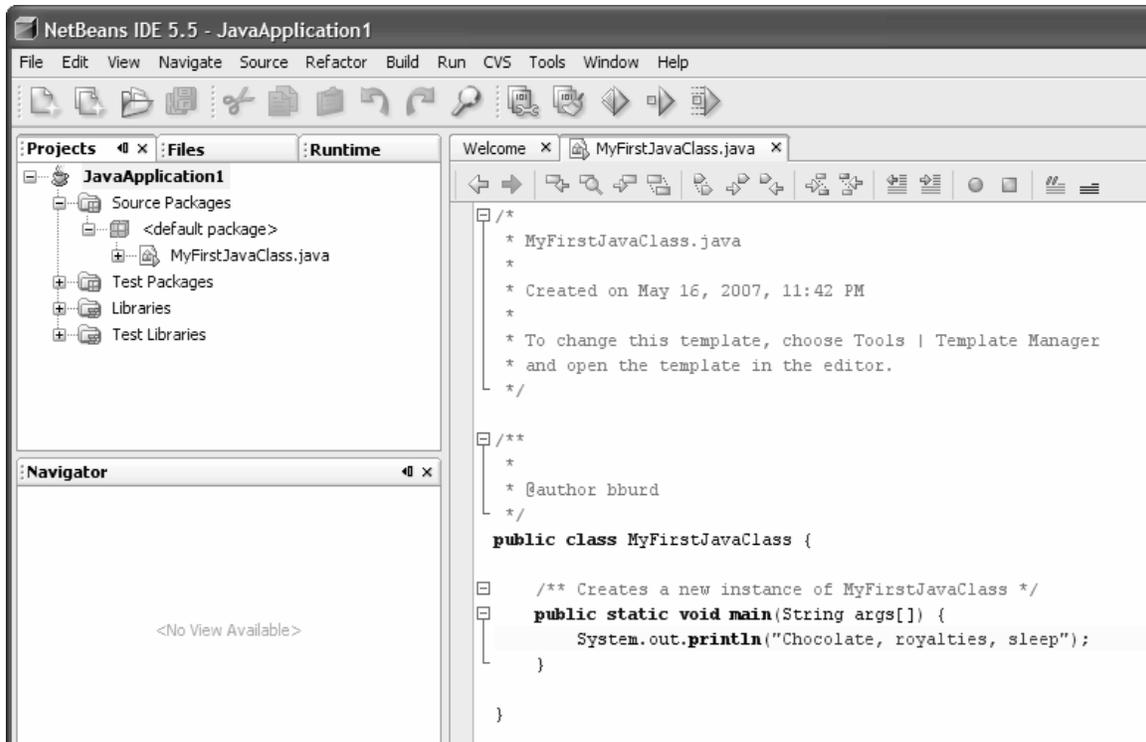
In this example, I type **MyFirstJavaClass**, because I intend to run the code in Listing 3-1. The name of the Java source file is `MyFirstJavaClass.java`, but in the Class Name field, I type **MyFirstJavaClass**, without the `.java` extension.

11. Click Finish.

The new class appears in the Projects window. Some skeletal code for the new class appears in the editor.



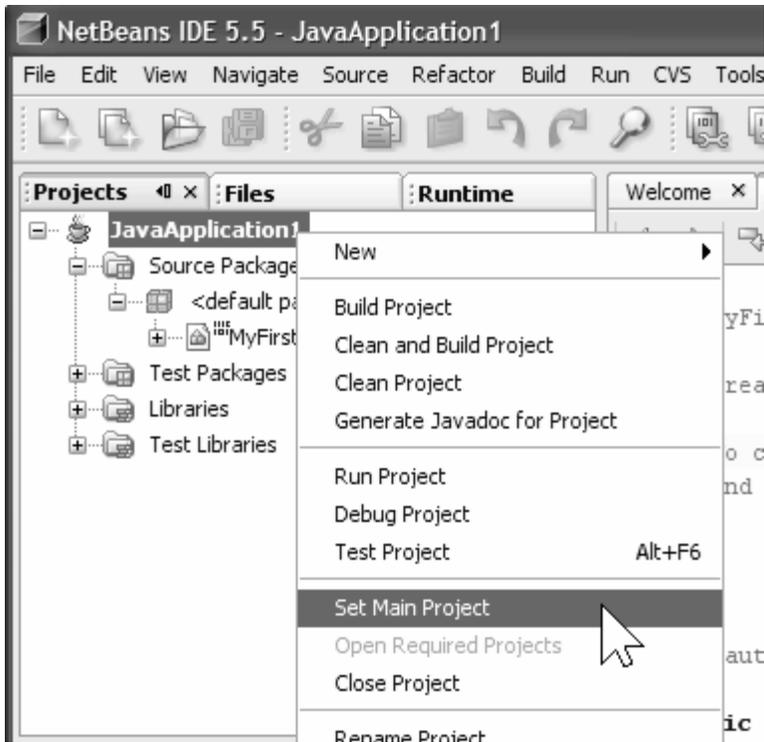
12. If you're creating code from scratch, you can type additional code in the editor. But if you're running an example from the book, copy that example's code, and paste the code into the NetBeans editor (replacing the skeletal code that's already in the editor).



13. On the main NetBeans menu, choose File → Save.

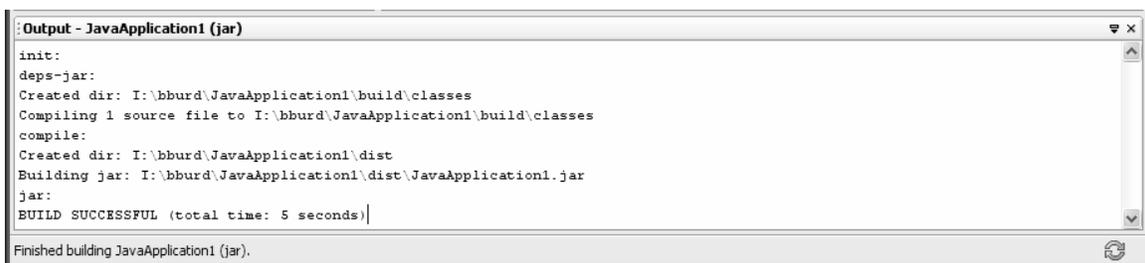
14. In the NetBeans Projects window, right-click the current project branch. In the resulting context menu, select Set Main Project.

On a Macintosh, substitute Ctrl-Click for right-click.



15. On the main NetBeans menu, choose Build → Build Main Project.

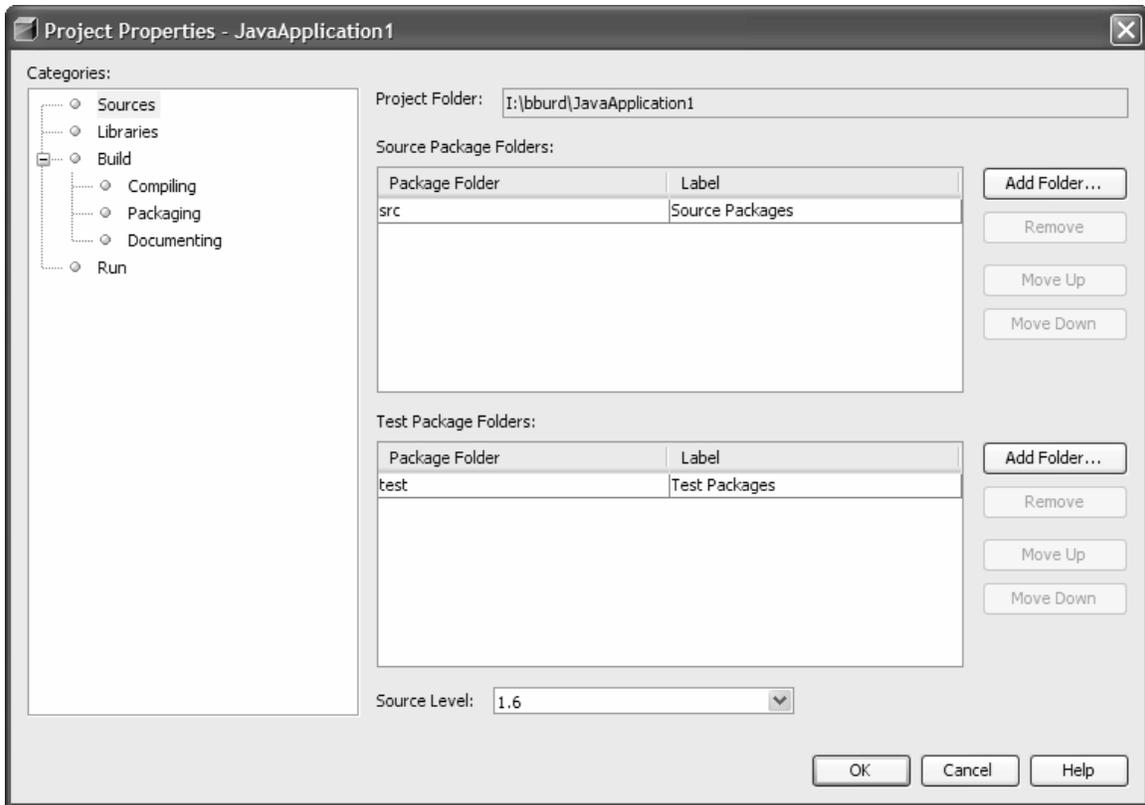
The output window shows a successful compilation.



If the compilation isn't successful, make changes to the code in the editor window, and try compiling again.

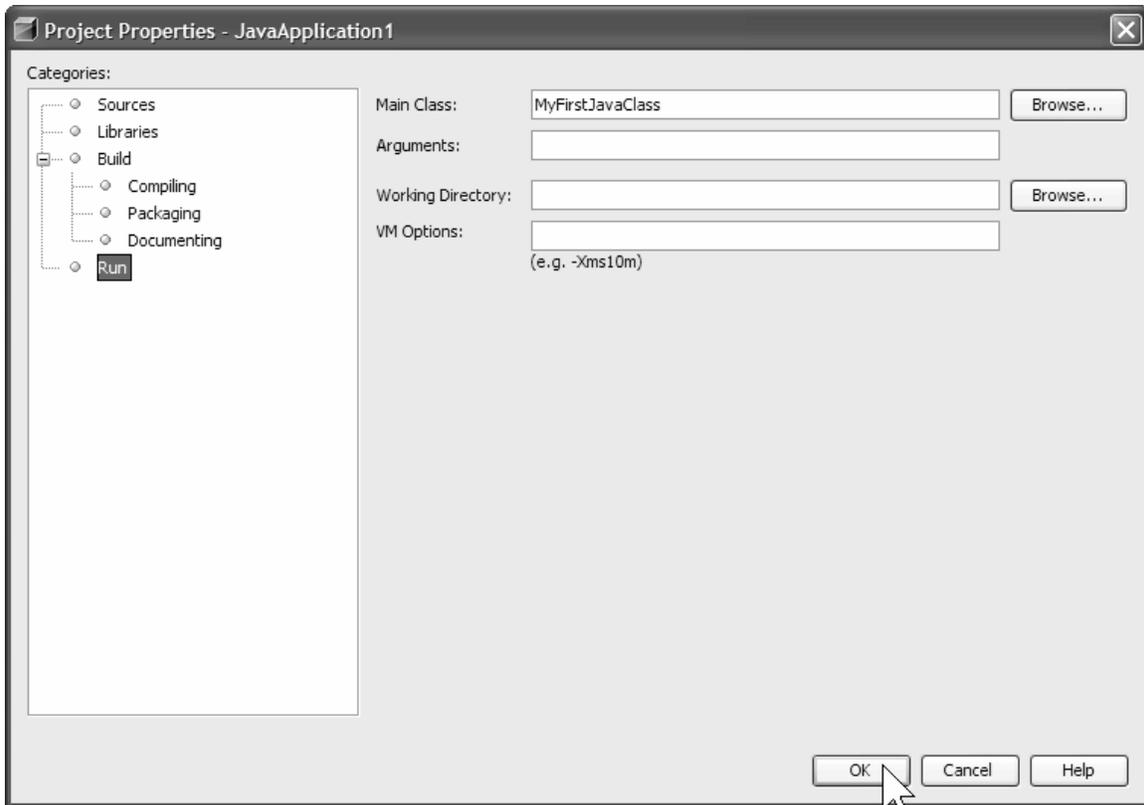
16. In the Projects window, right-click the current project's branch. In the resulting context menu, select Properties.

(On a Macintosh, substitute Ctrl-Click for right-click.) The Project Properties dialog appears.



17. On the left side of the Project Properties dialog, select Run.

The fields on the right side change to include Main Class, Arguments, and so on.



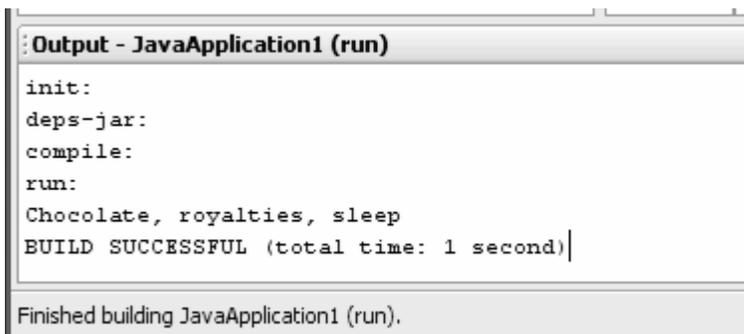
18. In the Main Class field, type the name of the class that contains the main method.

In this example, I type **MyFirstJavaClass**, because I intend to run the code in Listing 3-1. The name of the Java source file is `MyFirstJavaClass.java`, but in the Main Class field, I type **MyFirstJavaClass**, without the `.java` extension.

19. Click OK to dismiss the Project Properties dialog.

20. On the main NetBeans menu, choose Run → Run Main Project.

The program's output appears in the NetBeans Output window, and everybody is happy.

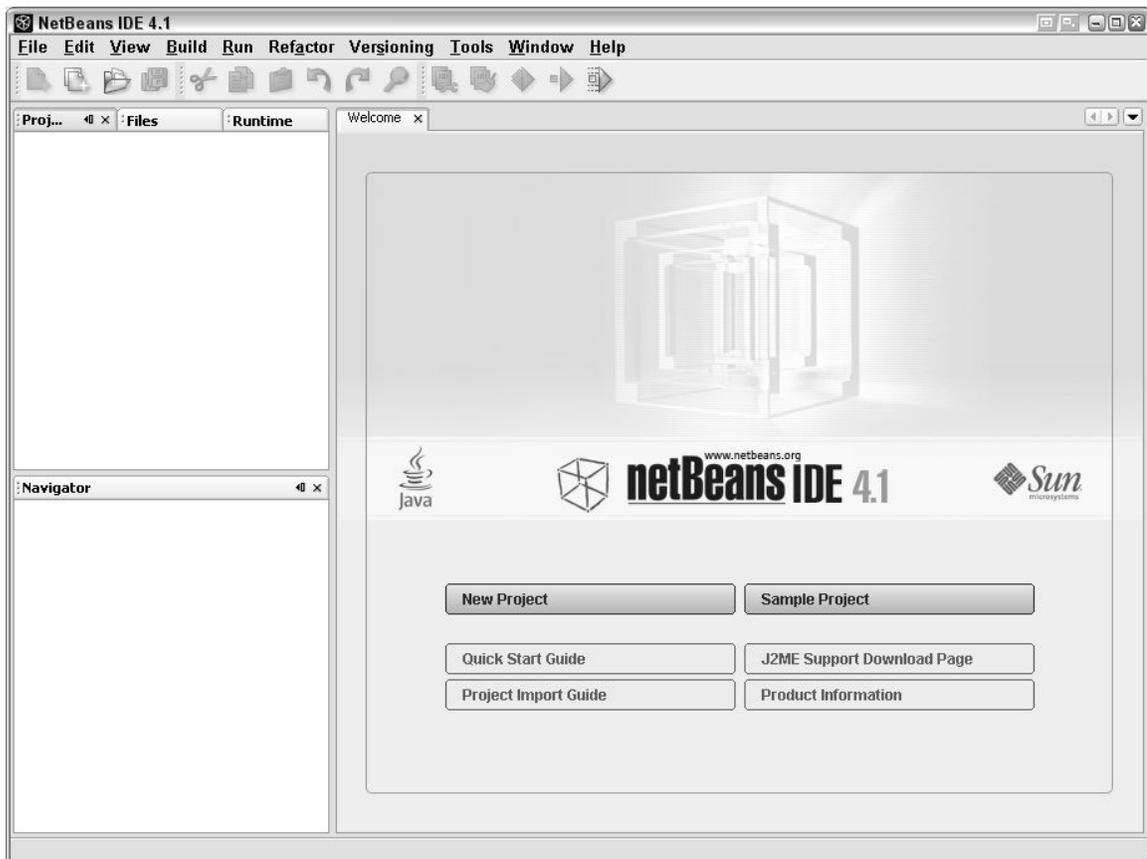


NetBeans 4.1

1. Download NetBeans by visiting <http://www.netbeans.org>.
2. Install the downloaded NetBeans program on your computer.
3. Run NetBeans by clicking the NetBeans icon.

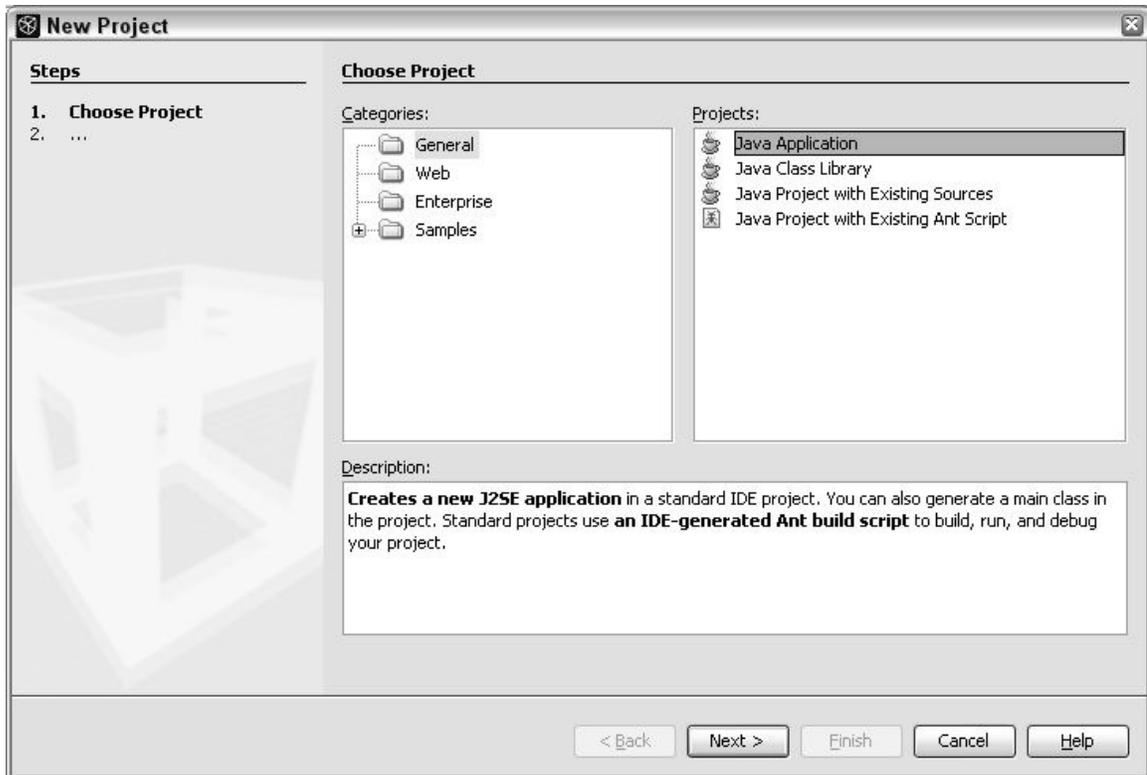


You see the NetBeans 4.1 opening screen.



4. Click New Project.

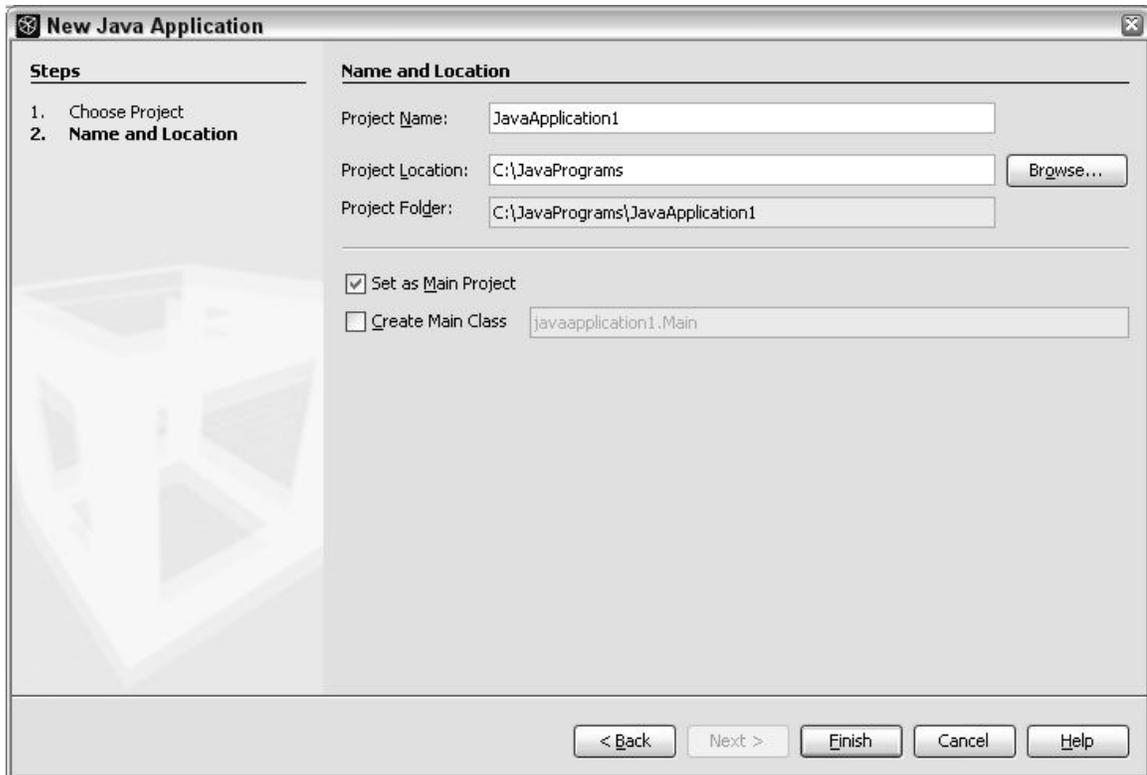
The New Project dialog appears.



5. In the Categories pane, select General, and in the Projects pane, select Java Application.

6. Click Next.

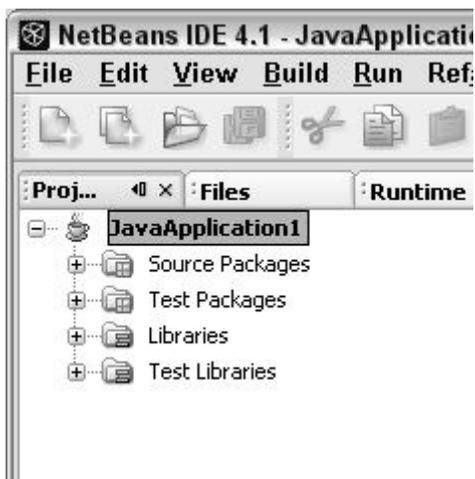
The New Java Application dialog appears.



7. Unless you're fussy, you can accept whatever defaults you see in the Project Name, Project Location, and Project Folder fields. Leave Set as Main Project checked. But in this example, you should uncheck Create Main Class.

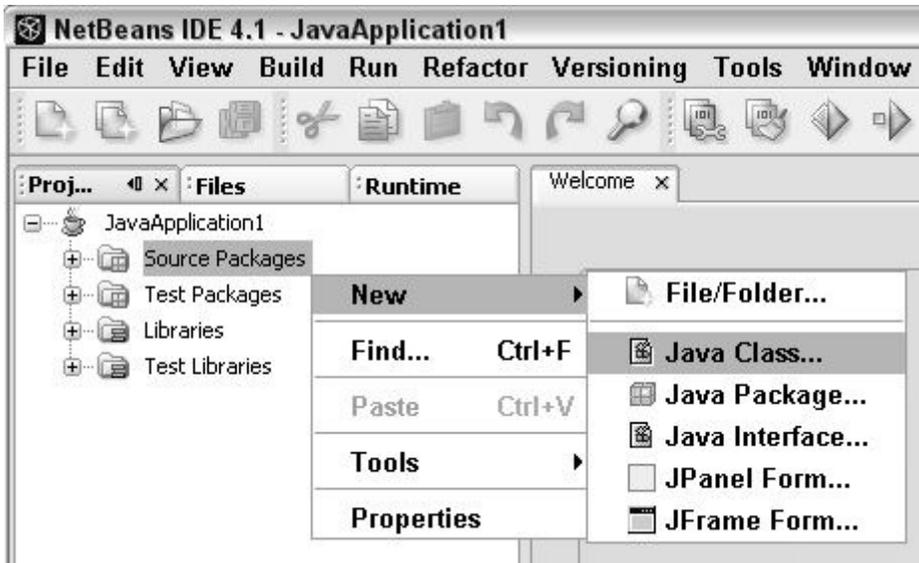
8. Click Finish.

A new project tree appears in the NetBeans Projects window.

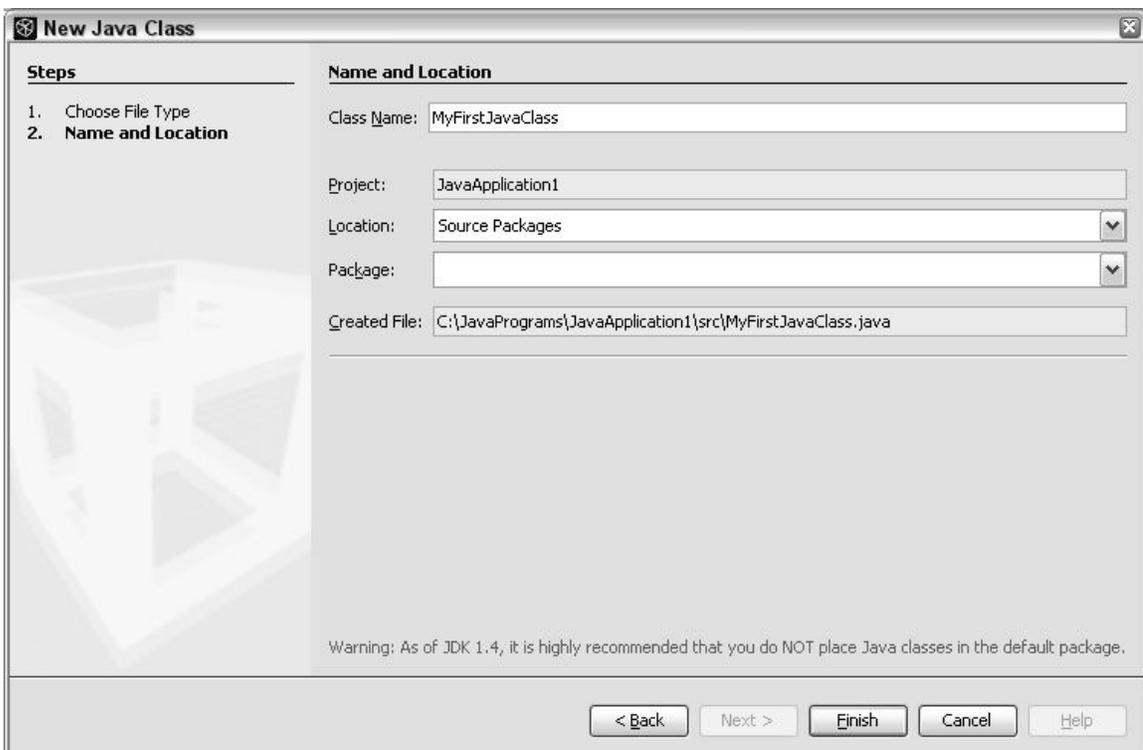


9. On the project tree, right-click the Source Package branch. In the resulting context menus, select New → Java Class.

On a Macintosh, substitute Ctrl-Click for right-click.



The New Java Class dialog appears.

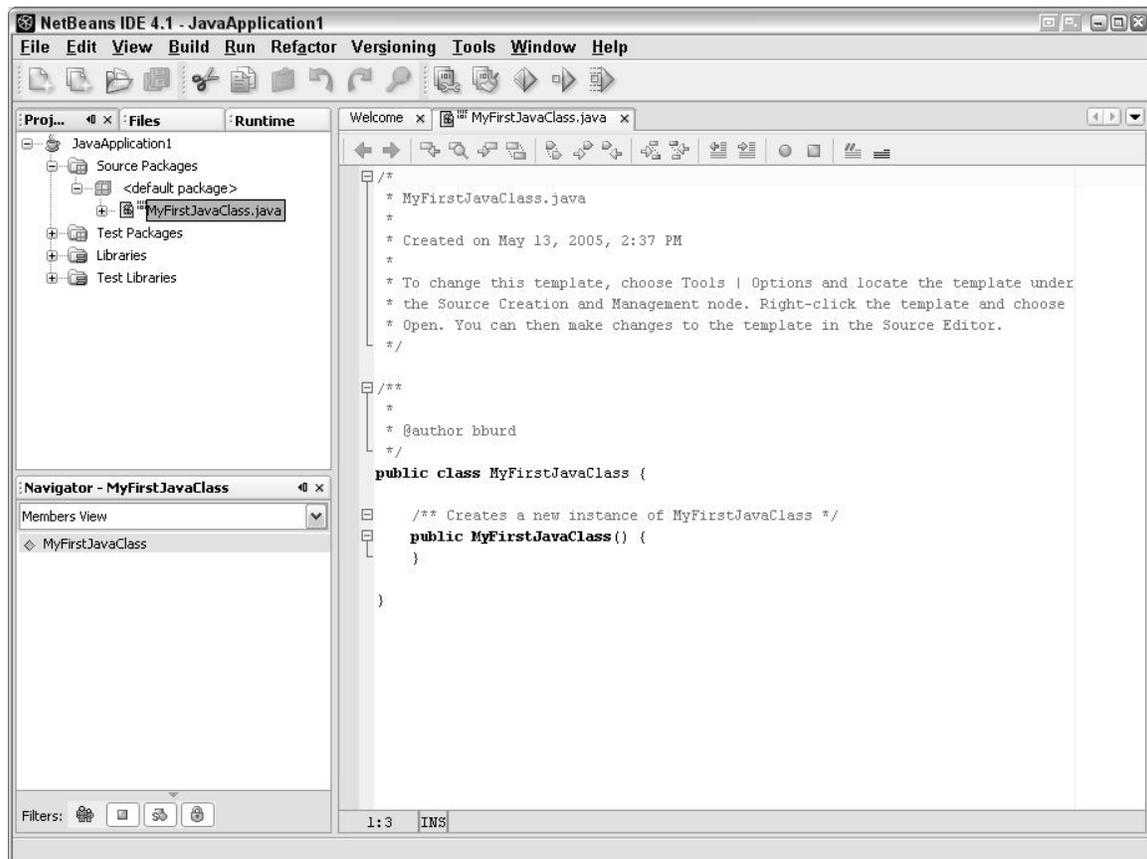


10. In the Class Name field, type the name of the class that you want to create.

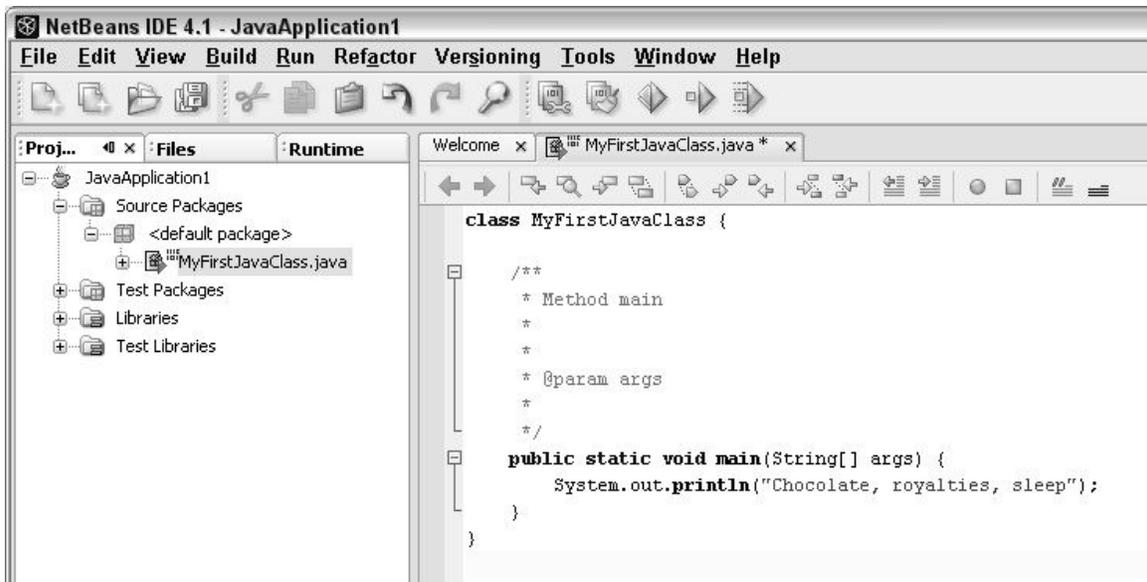
In this example, I type **MyFirstJavaClass**, because I intend to run the code in Listing 3-1. The name of the Java source file is `MyFirstJavaClass.java`, but in the Class Name field, I type **MyFirstJavaClass**, without the `.java` extension.

11. Click Finish.

The new class appears in the Projects window. Some skeletal code for the new class appears in the editor.



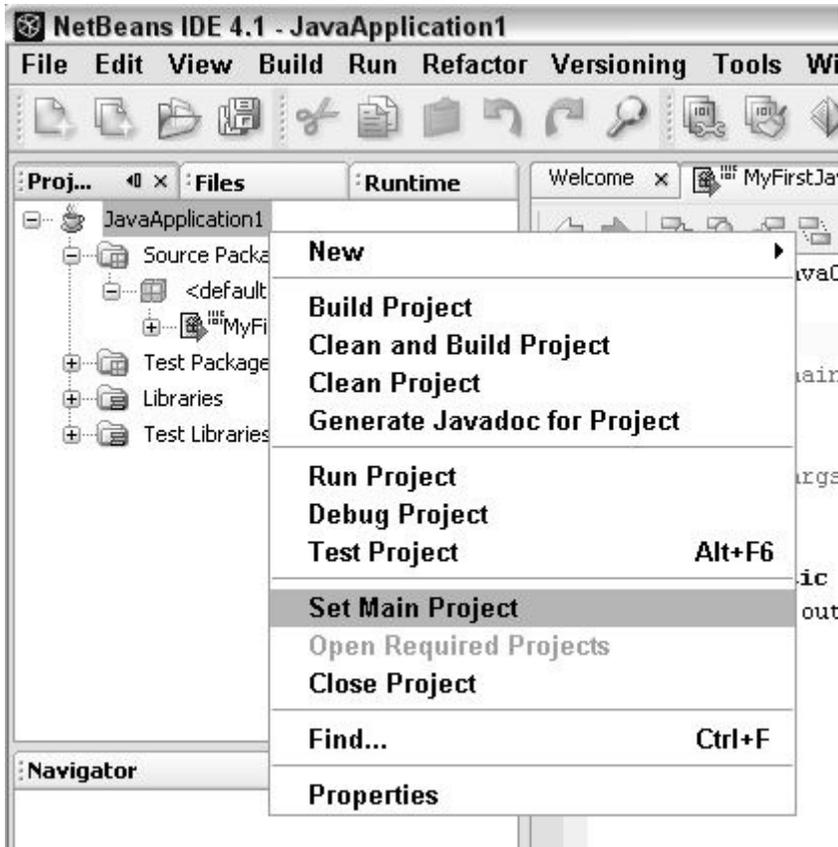
12. If you're creating code from scratch, you can type additional code in the editor. But if you're running an example from the book, copy that example's code, and paste the code into the NetBeans editor (replacing the skeletal code that's already in the editor).



13. On the main NetBeans menu, choose File → Save.

14. In the NetBeans Projects window, right-click the current project branch. In the resulting context menu, select Set Main Project.

On a Macintosh, substitute Ctrl-Click for right-click.



15. On the main NetBeans menu, choose Build → Build Main Project.

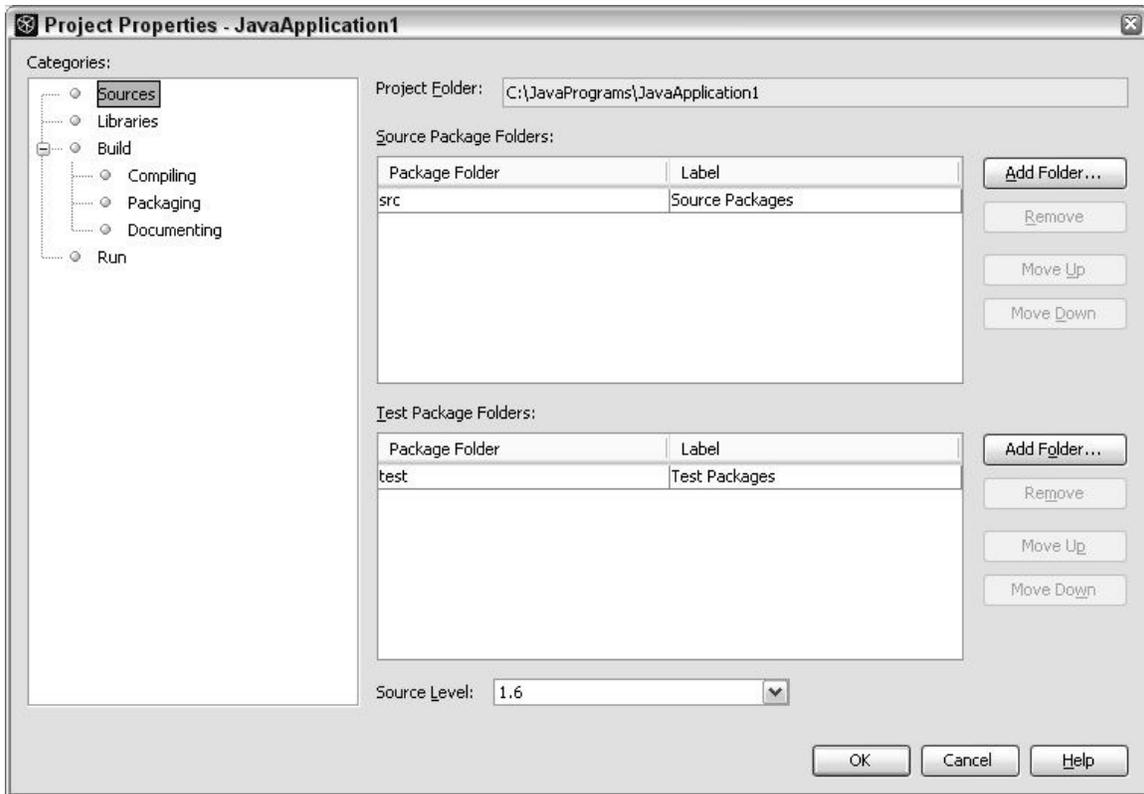
The output window shows a successful compilation.



If the compilation isn't successful, make changes to the code in the editor window, and try compiling again.

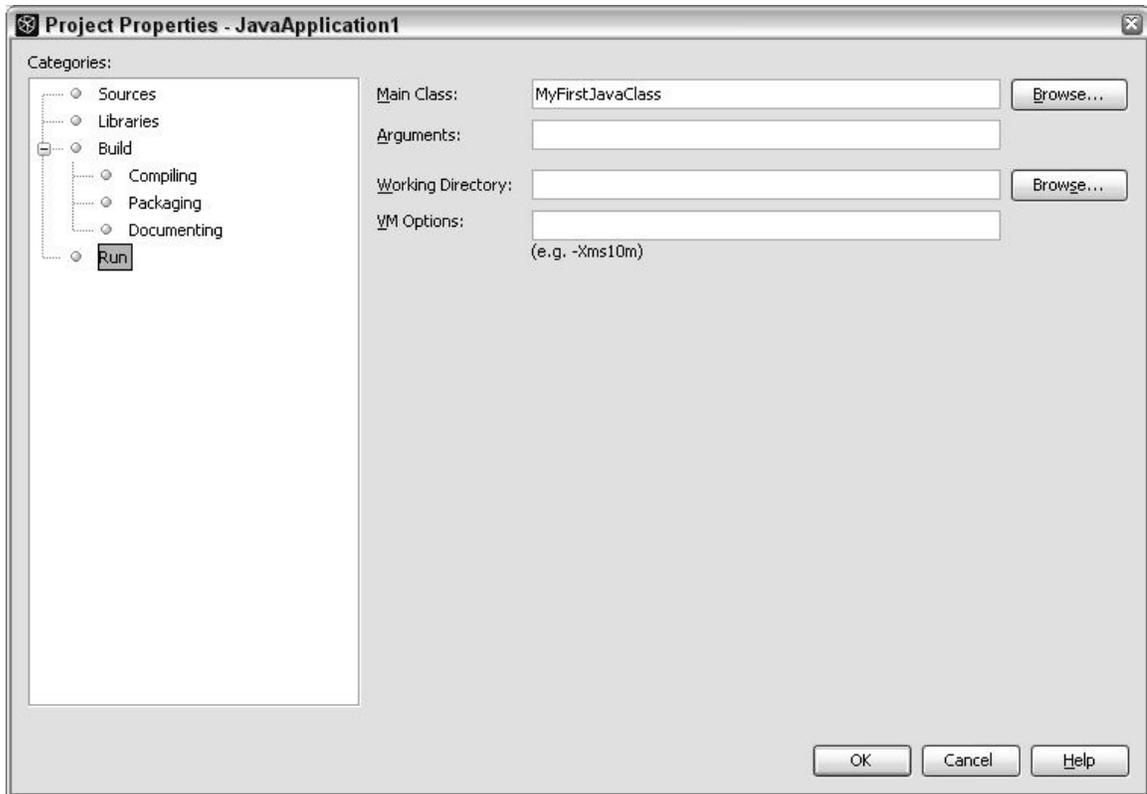
16. In the Projects window, right-click the current project's branch. In the resulting context menu, select Properties.

(On a Macintosh, substitute Ctrl-Click for right-click.) The Project Properties dialog appears.



17. On the left side of the Project Properties dialog, select Run.

The fields on the right side change to include Main Class, Arguments, and so on.



18. In the Main Class field, type the name of the class that contains the main method.

In this example, I type **MyFirstJavaClass**, because I intend to run the code in Listing 3-1. The name of the Java source file is `MyFirstJavaClass.java`, but in the Main Class field, I type **MyFirstJavaClass**, without the `.java` extension.

19. Click OK to dismiss the Project Properties dialog.

20. On the main NetBeans menu, choose Run → Run Main Project.

The program's output appears in the NetBeans Output window, and everybody is happy.

```
Output - JavaApplication1 (run)
init:
deps-jar:
compile:
run:
Chocolate, royalties, sleep
BUILD SUCCESSFUL (total time: 0 seconds)
```