3. NetBeans IDE 6.0

Java

Fall 2009

Instructor: Dr. Masoud Yaghini

Outline

- Installing the NetBeans IDE
- First NetBeans IDE Project
- IDE Windows
- Source Editor
- Customizing the IDE
- References

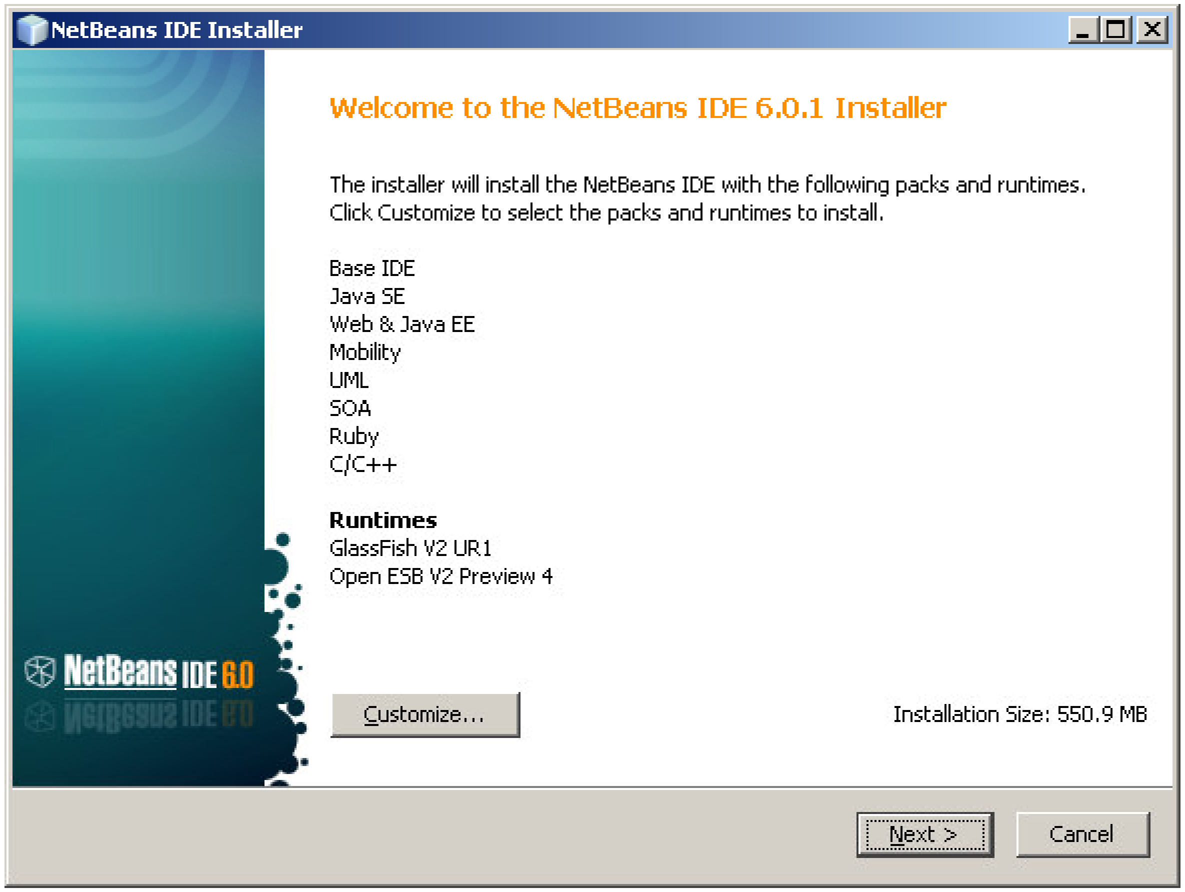
Installing the NetBeans IDE

Installing the NetBeans IDE

1) Execute the installer
   - you will see the NetBeans 6 welcome page

Installing the NetBeans IDE

2) Click the Customize button to select which features you want to install



Welcome to the NetBeans IDE 6.0.1 Installer

The installer will install the NetBeans IDE with the following packs and runtimes.
Click Customize to select the packs and runtimes to install.

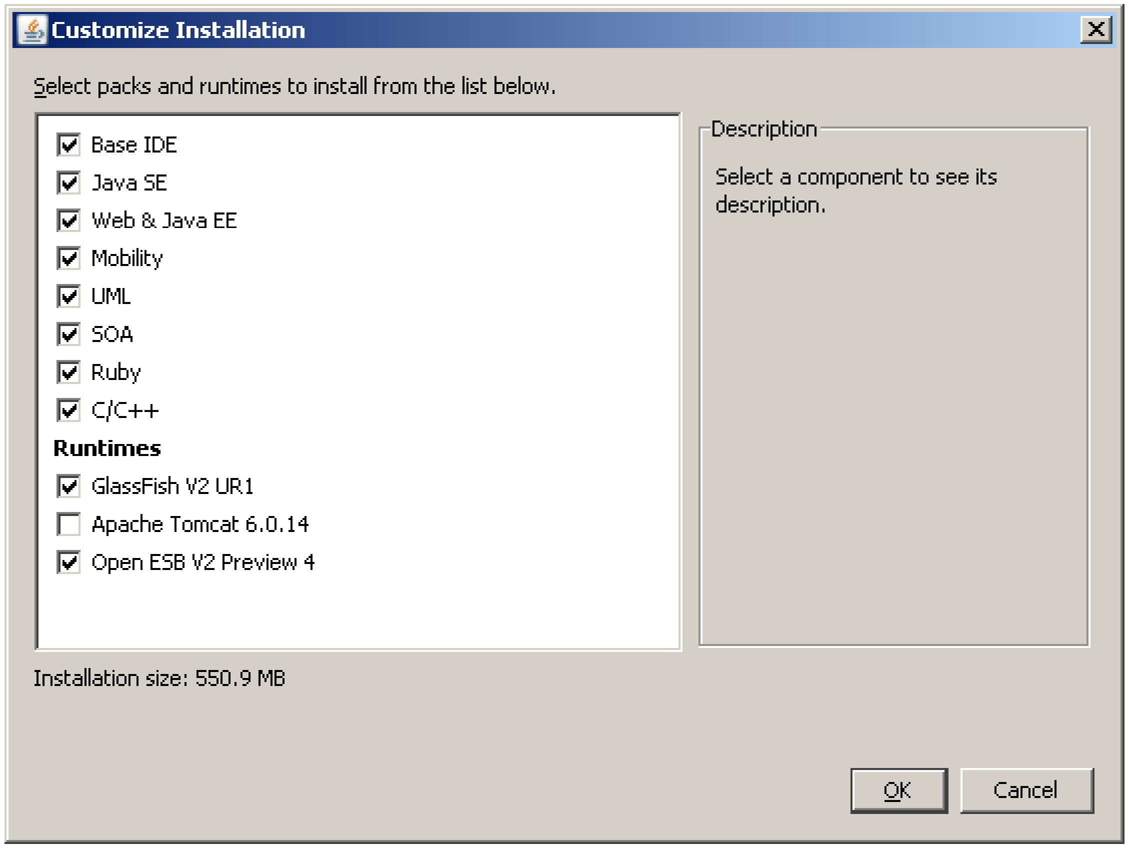
Base IDE
Java SE
Web & Java EE
Mobility
UML
SOA
Ruby
C/C++

Runtimes
GlassFish V2 UR1
Open ESB V2 Preview 4

Installation Size: 550.9 MB

Installing the NetBeans IDE

3) Select Base IDE, Java SE, and UML, and click OK


Customize Installation

Select packs and runtimes to install from the list below.

- [✓] Base IDE
- [✓] Java SE
- [✓] Web & Java EE
- [✓] Mobility
- [✓] UML
- [✓] SOA
- [✓] Ruby
- [✓] C/C++

Runtimes

- [✓] GlassFish V2 UR1
- [ ] Apache Tomcat 6.0.14
- [✓] Open ESB V2 Preview 4

Description

Select a component to see its description.

Installation size: 550.9 MB

Installing the NetBeans IDE

4) Select the check box next to the text “I Accept the terms in the license agreement” and click the Next button.

Installing the NetBeans IDE

5) Next step you should select installation folders for :
   - the NetBeans IDE
   - the JDK

- By default:
  - C:\Program Files\NetBeans 6.0.1
  - C:\Program Files\Java\jdk1.6.0_06
- Click the Browse button and select a directory for change the default directories

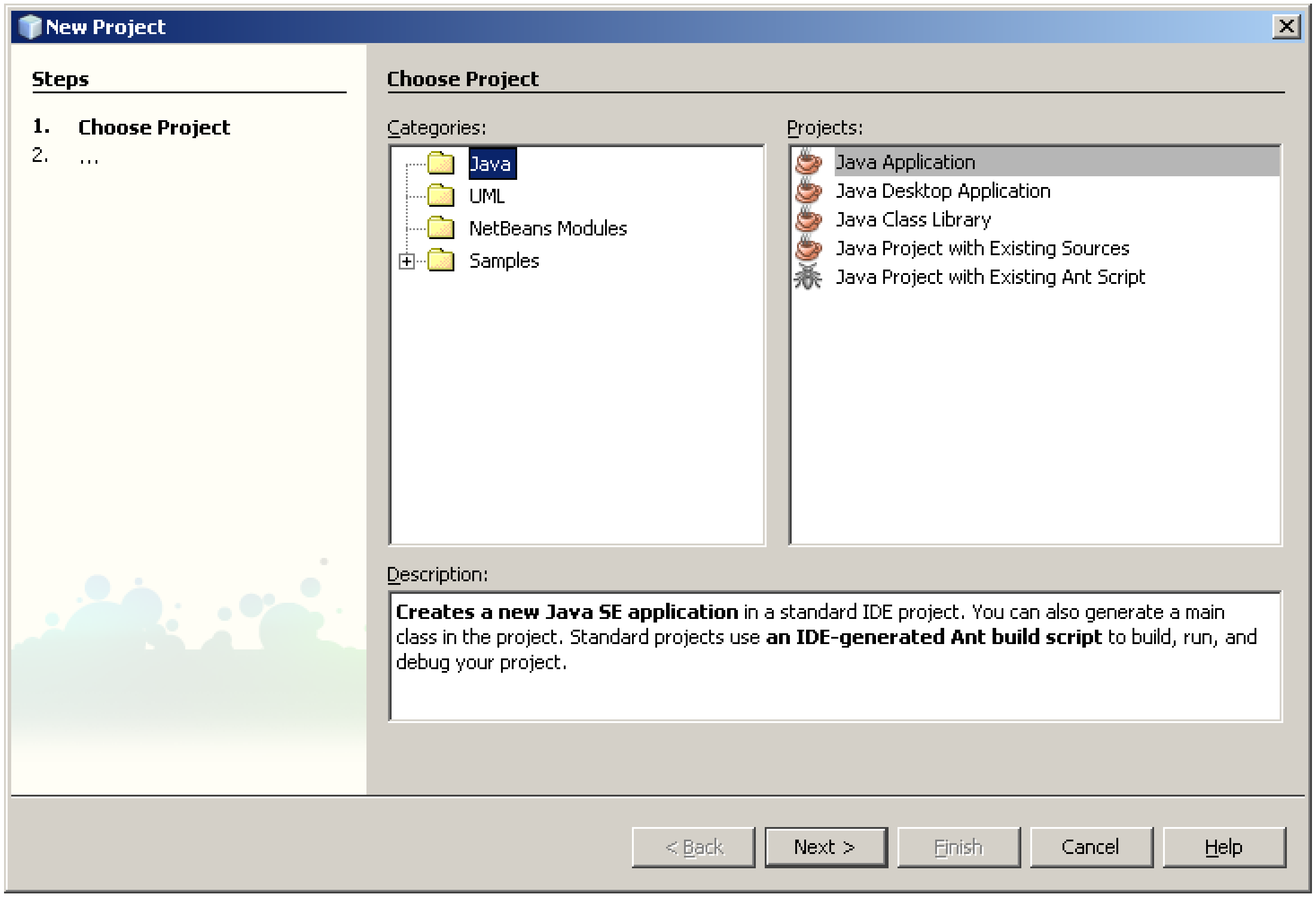
Installing the NetBeans IDE

6) After selecting directories, click Finish for staring installation

First NetBeans IDE Project

First NetBeans IDE Project

1) Choose File | New Project


New Project

Steps

1. Choose Project
2. ...

Choose Project

Categories:
- Java
- UML
- NetBeans Modules
- Samples

Projects:
- Java Application
- Java Desktop Application
- Java Class Library
- Java Project with Existing Sources
- Java Project with Existing Ant Script

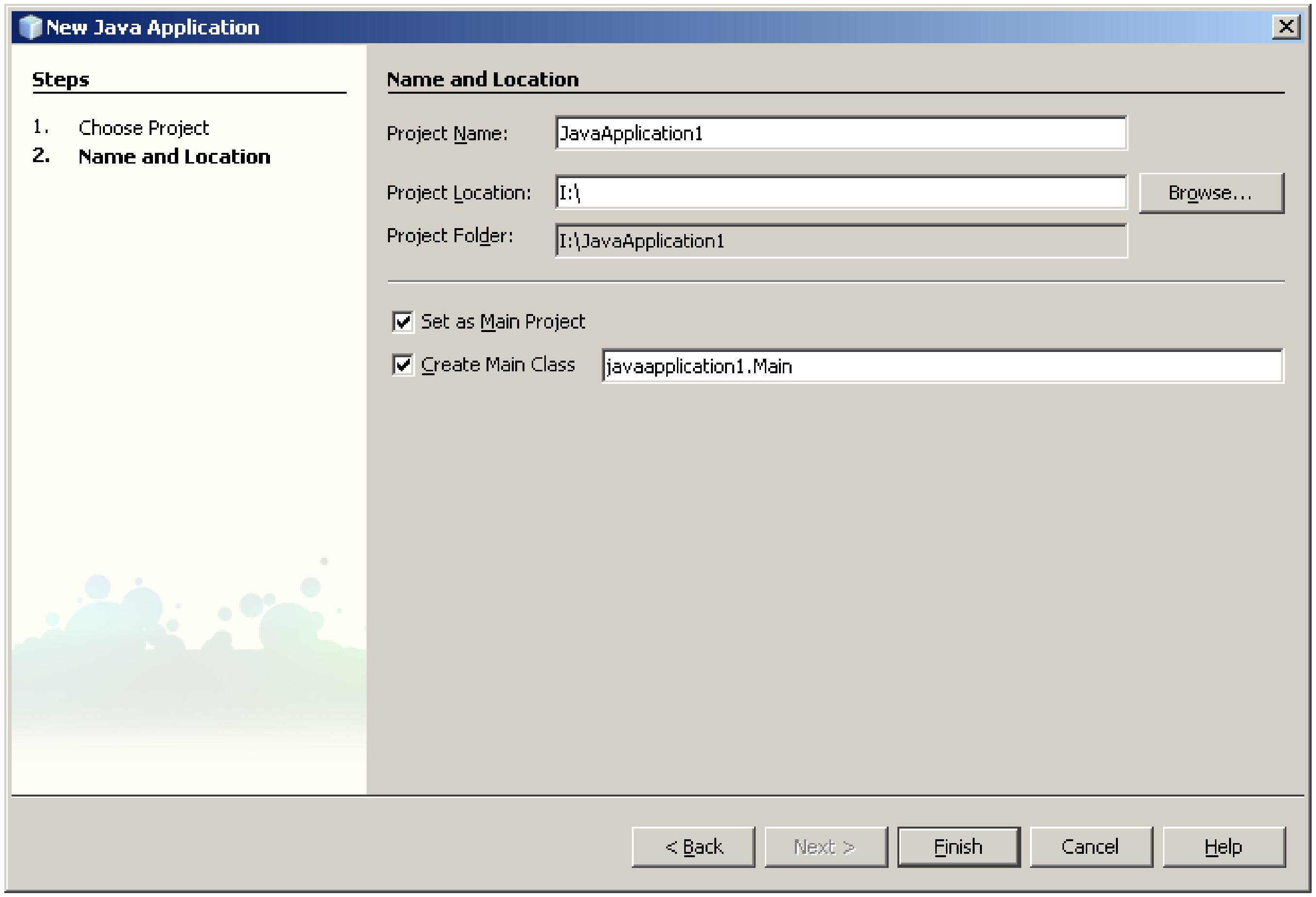
Description:

Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

< Back | Next > | Finish | Cancel | Help

First NetBeans IDE Project

2) In the New Project wizard, select Java Application, and click Next


New Java Application

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

Name and Location

Project Name: JavaApplication1

Project Location: I:\

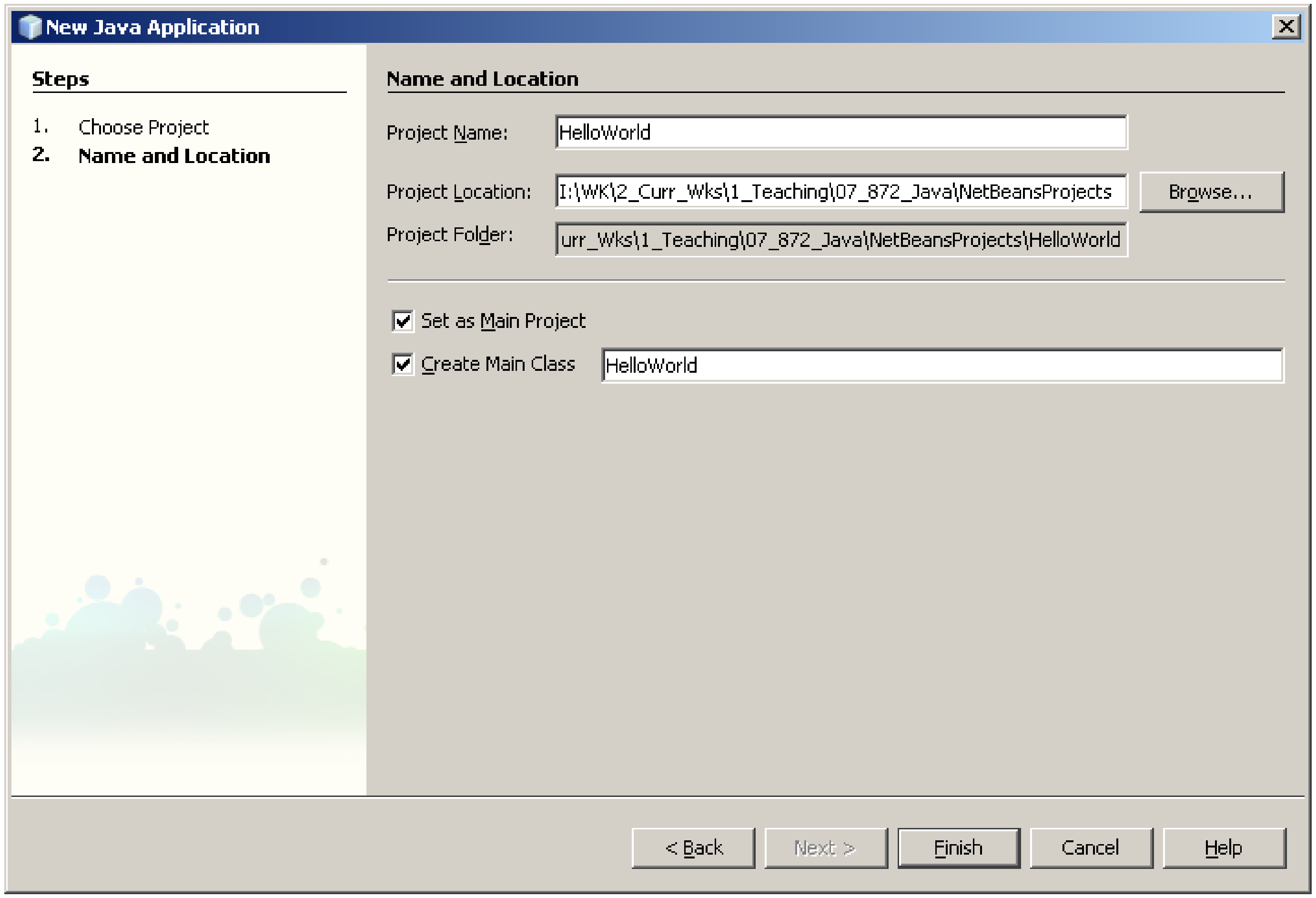
Project Folder: I:\JavaApplication1

[✓] Set as Main Project

[✓] Create Main Class javaapplication1.Main

First NetBeans IDE Project

3) In the Project Name type HelloWorld, In the Project Location choose the suitable directory, In the Create Main Class type HelloWorld


New Java Application

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

Name and Location

Project Name: HelloWorld

Project Location: I:\WK\2_Curr_Wks\1_Teaching\07_872_Java\NetBeansProjects

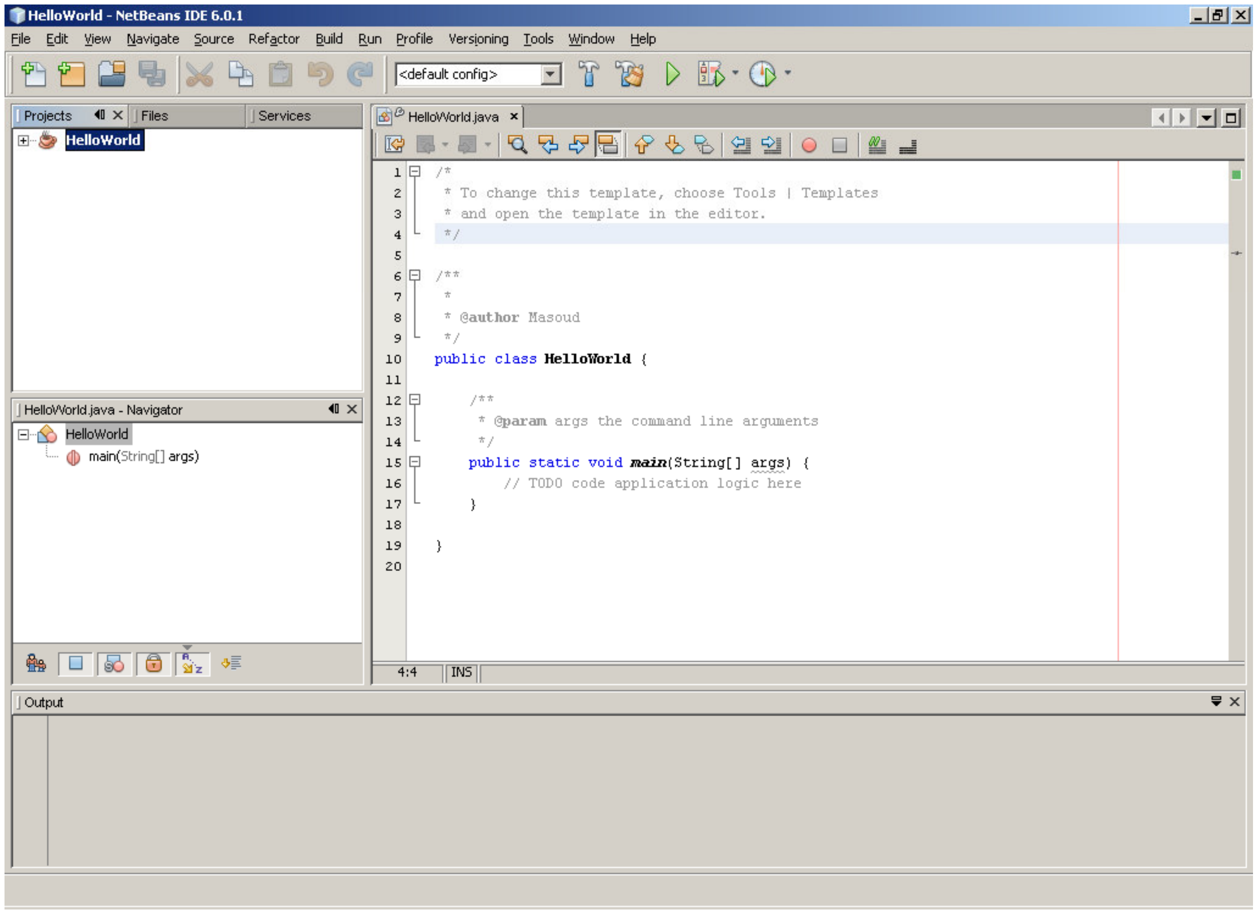
Project Folder: urr_Wks\1_Teaching\07_872_Java\NetBeansProjects\HelloWorld

[✓] Set as Main Project

[✓] Create Main Class: HelloWorld

First NetBeans IDE Project

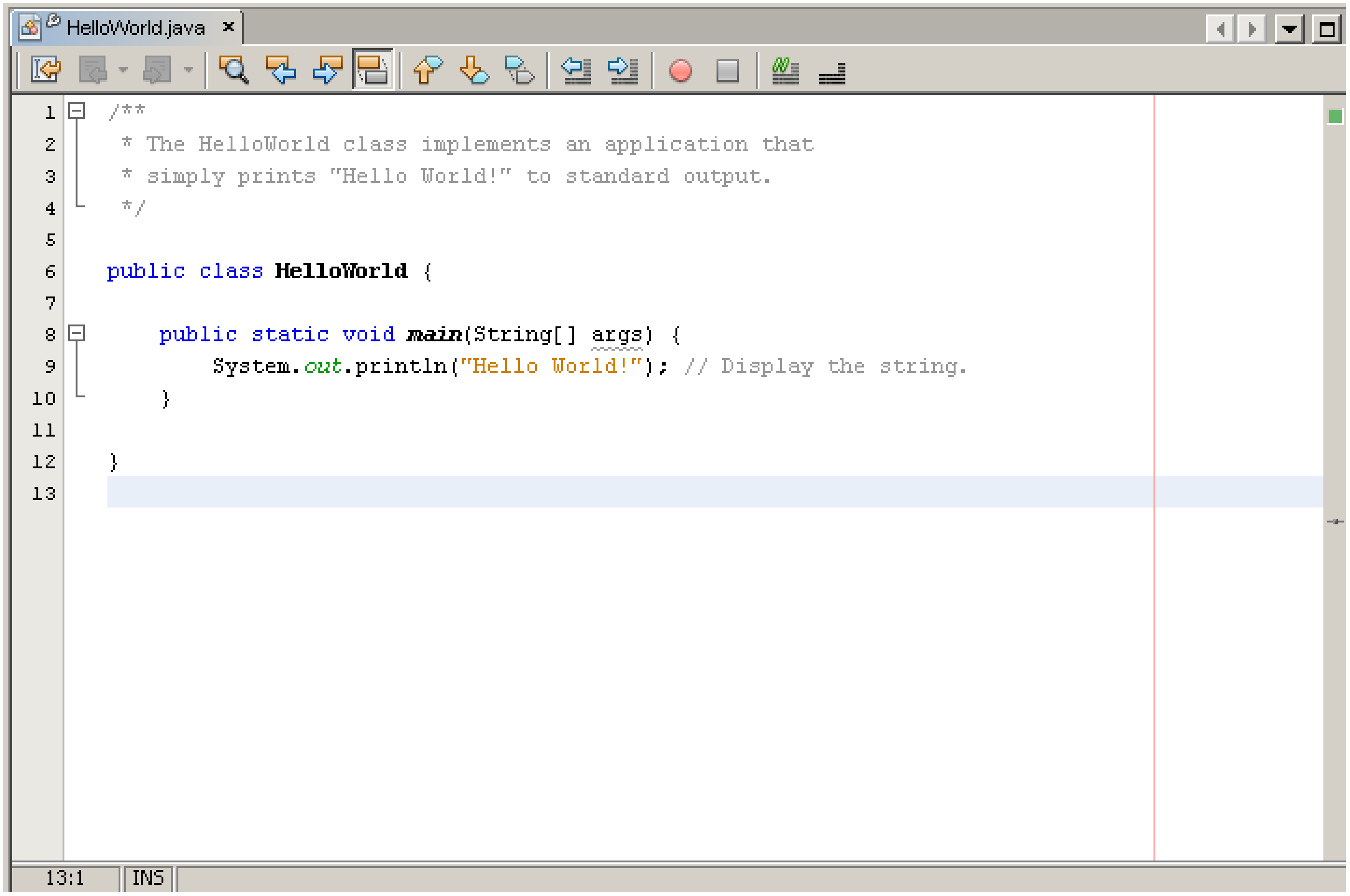
4) Click Finish


HelloWorld - NetBeans IDE 6.0.1
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
Files
Services
HelloWorld
HelloWorld.java - Navigator
HelloWorld
main(String[] args)
HelloWorld.java
4:4
INS
Output

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Masoud
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

First NetBeans IDE Project

6) In the Source Editor, type the comment and command of HelloWorld program

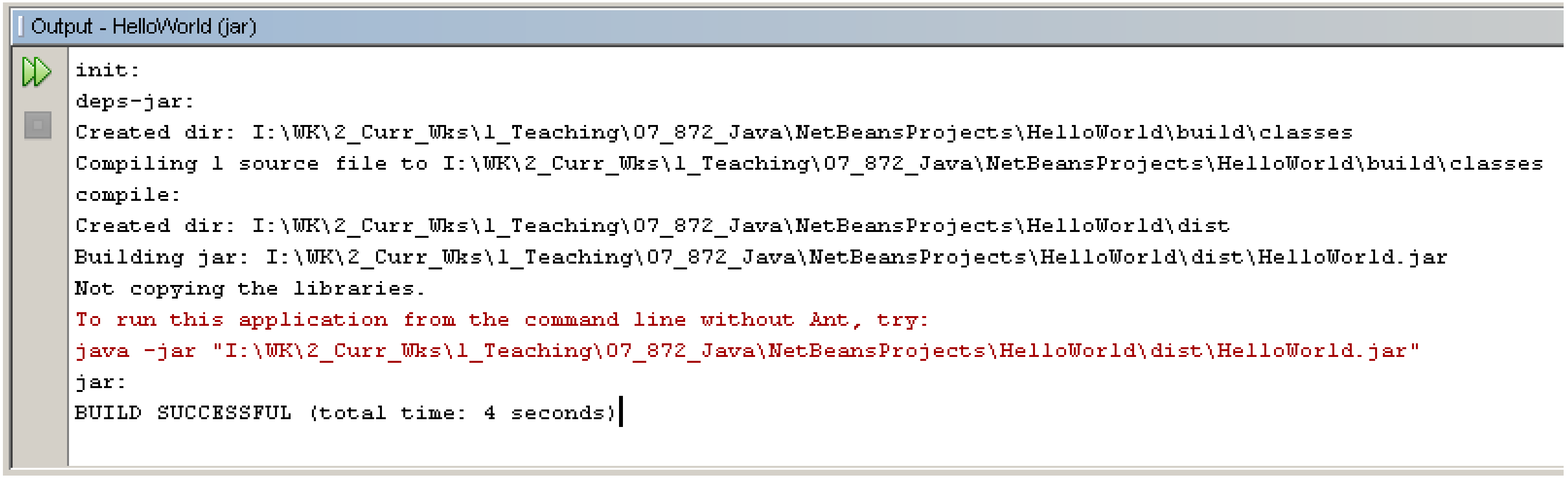


```
/**
 * The HelloWorld class implements an application that
 * simply prints "Hello World!" to standard output.
 */

public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }

}
```

First NetBeans IDE Project

7) Press Ctrl-S to save the application.

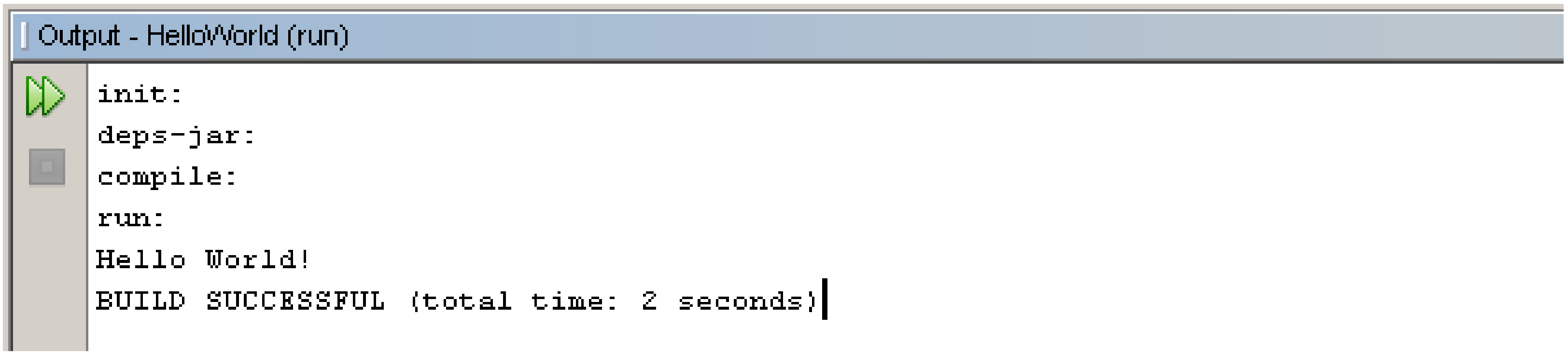
8) Press F11 (or choose Build | Build Main Project) to compile the application. The Output window opens and displays the output



```
init:
deps-jar:
Created dir: I:\WK\2_Curr_Wks\1_Teaching\07_872_Java\NetBeansProjects\HelloWorld\build\classes
Compiling 1 source file to I:\WK\2_Curr_Wks\1_Teaching\07_872_Java\NetBeansProjects\HelloWorld\build\classes
compile:
Created dir: I:\WK\2_Curr_Wks\1_Teaching\07_872_Java\NetBeansProjects\HelloWorld\dist
Building jar: I:\WK\2_Curr_Wks\1_Teaching\07_872_Java\NetBeansProjects\HelloWorld\dist\HelloWorld.jar
Not copying the libraries.
To run this application from the command line without Ant, try:
java -jar "I:\WK\2_Curr_Wks\1_Teaching\07_872_Java\NetBeansProjects\HelloWorld\dist\HelloWorld.jar"
jar:
BUILD SUCCESSFUL (total time: 4 seconds)
```

First NetBeans IDE Project

9) Press F6 (or choose Run | Run Main Project) to run the project.



Output - HelloWorld (run)

```
init:
deps-jar:
compile:
run:
Hello World!
BUILD SUCCESSFUL (total time: 2 seconds)
```

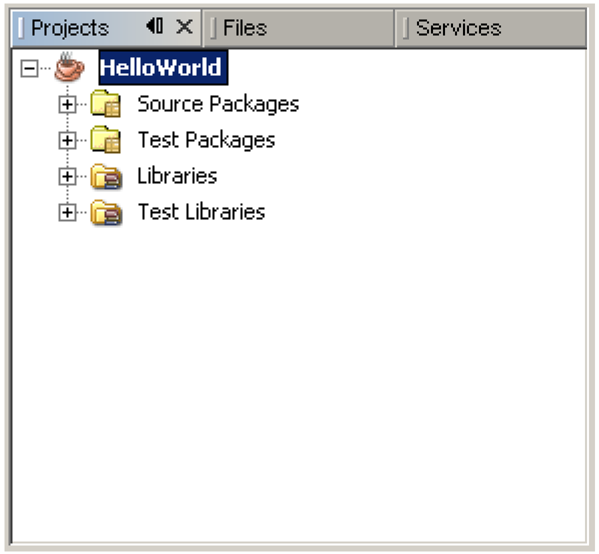
IDE Windows

IDE Windows

- There are multiple windows you can open and use throughout the IDE windowing system.
- Each window has a specific purpose and can be opened, minimized, or closed.
- You can choose each of window by selecting Window

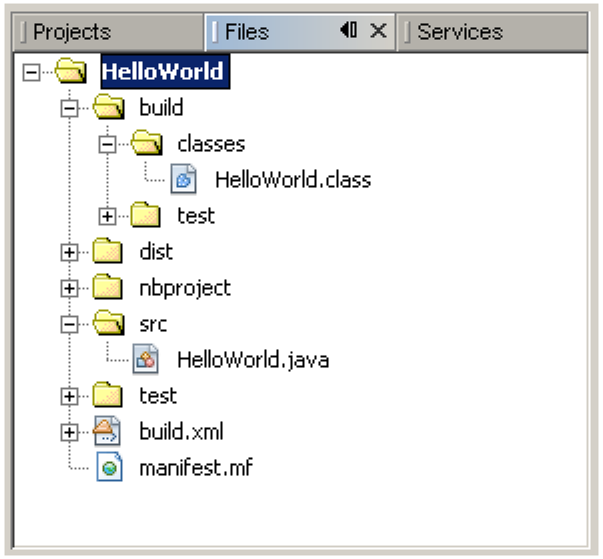
IDE Windows

- **Projects Window**
  - The Projects window displays all the currently opened projects.
  - It is the main entry point for NetBeans to categorize and group files for use in an application.
  - For most Java project types, the files are sorted into four groups:
    - Source Packages
    - Test Packages
    - Libraries
    - Test Libraries



IDE Windows

- **Files Window**
  - The Files window provides a more normal file-based view of open projects.
  - The files in a project are organized in a folder-and-file structure



IDE Windows

- **Services Window**
  - The Services window is where you can find important resources such as HTTP servers, database servers, web services, DTD and XML schema catalogs, and processes.
- **Navigator Window**
  - The Navigator window provides a quick-and-easy view of a node that has been selected in the Projects window or Source Editor.
  - It can display the methods, constructors, and fields in a class in a traditional list view or as an inheritance tree.

IDE Windows

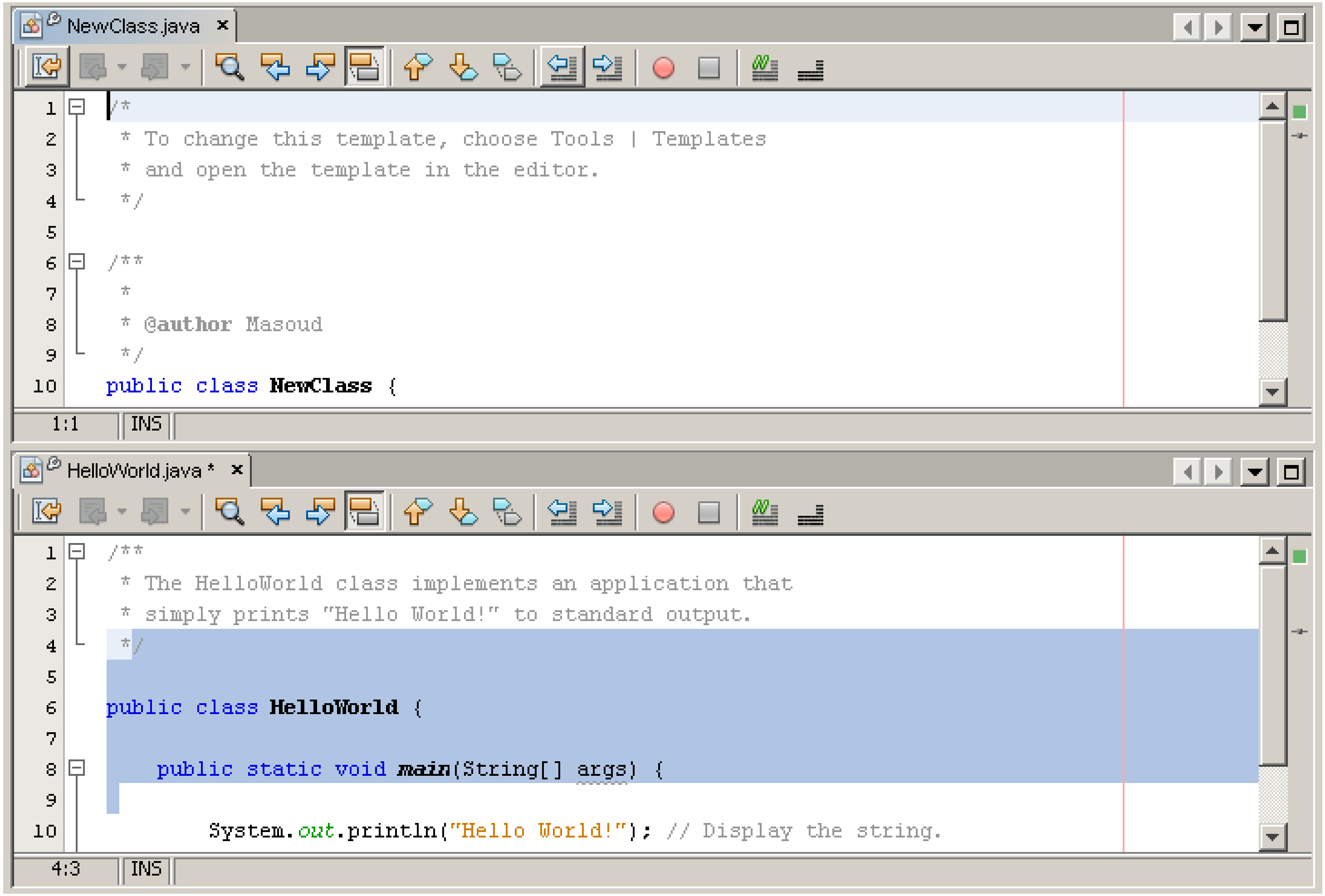
- **Source Editor**
  - The Source Editor window is where you edit code and other files.
  - When you open files, they appear in the Source Editor window as a tabbed view.
- **Output Window**
  - If you choose to build your project, compile a single file, or run a file that outputs text to the standard output or standard error stream, the information and results are displayed in the Output window.

Source Editor

Arranging Files in the Source Editor

- The Source Editor allows you to arrange files in many different ways
- The default viewing option
  - All the files open in the same window with the names of each file appearing in a tab.
- Dual-file editing
  - Click and hold the Filename tab for the second file, and move your mouse to the lower half of or right half of the Source Editor
- View the same file in two places
  - Right-clicking the Filename tab and selecting Clone Document.

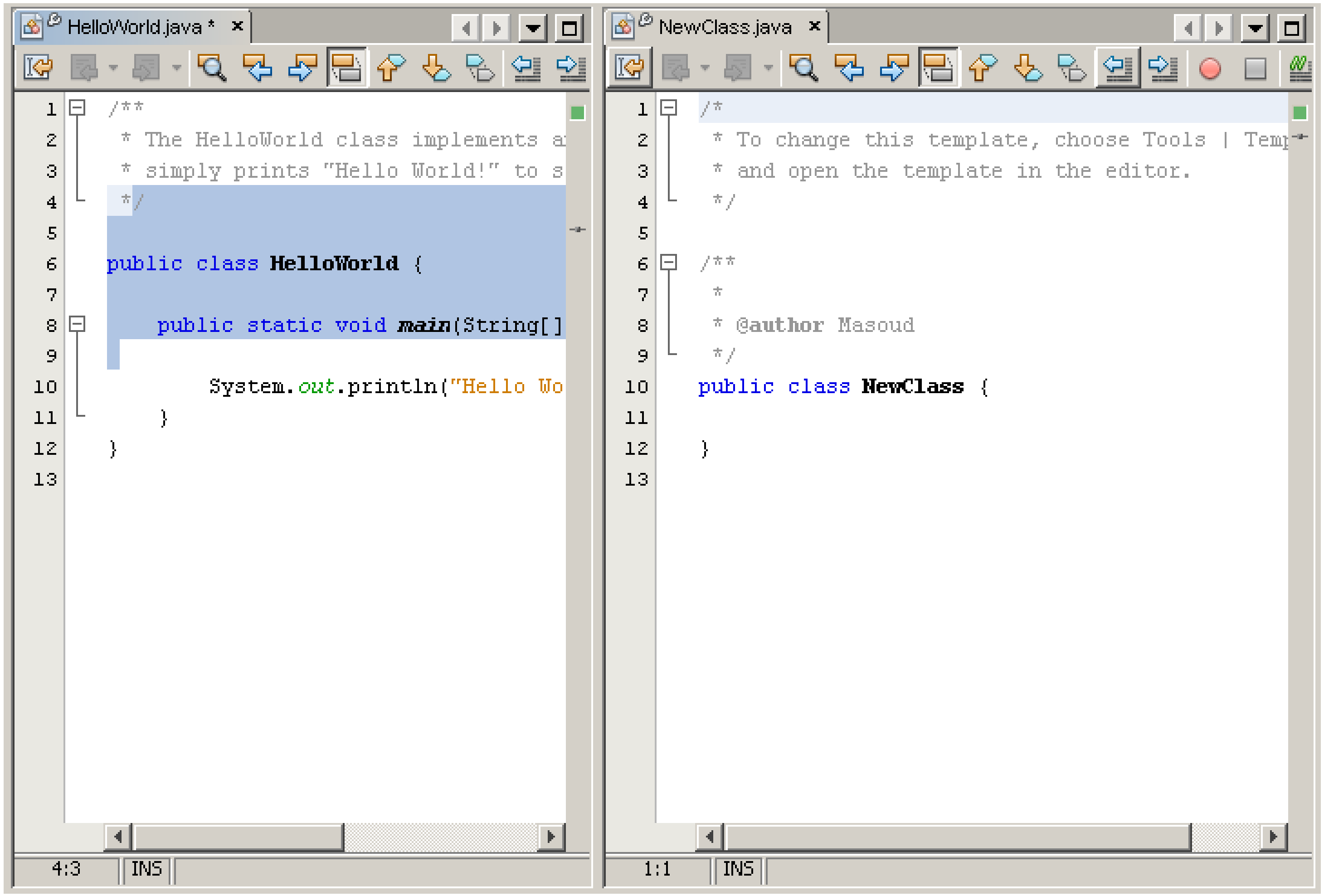
Dual-File Editing



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Masoud
 */
public class NewClass {
```

```
/**
 * The HelloWorld class implements an application that
 * simply prints "Hello World!" to standard output.
 */

public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello World!"); // Display the string.
```

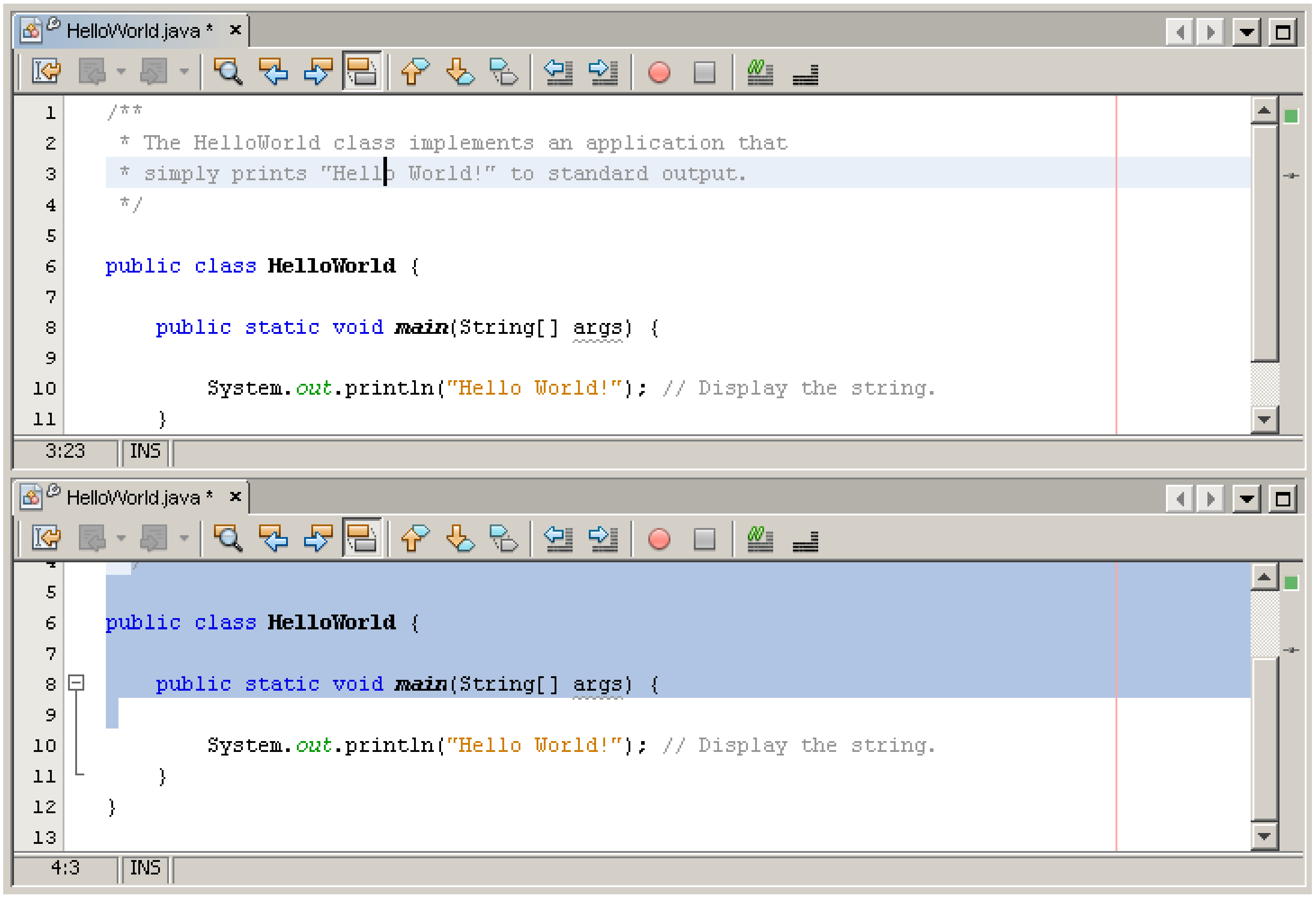
Dual-File Editing



```
/**
 * The HelloWorld class implements a
 * simply prints "Hello World!" to s
 */

public class HelloWorld {

    public static void main(String[]

        System.out.println("Hello Wo
    }
}
```

```
/*
 * To change this template, choose Tools | Temp
 * and open the template in the editor.
 */

/**
 *
 * @author Masoud
 */
public class NewClass {

}
```

View the Same File in Two Places

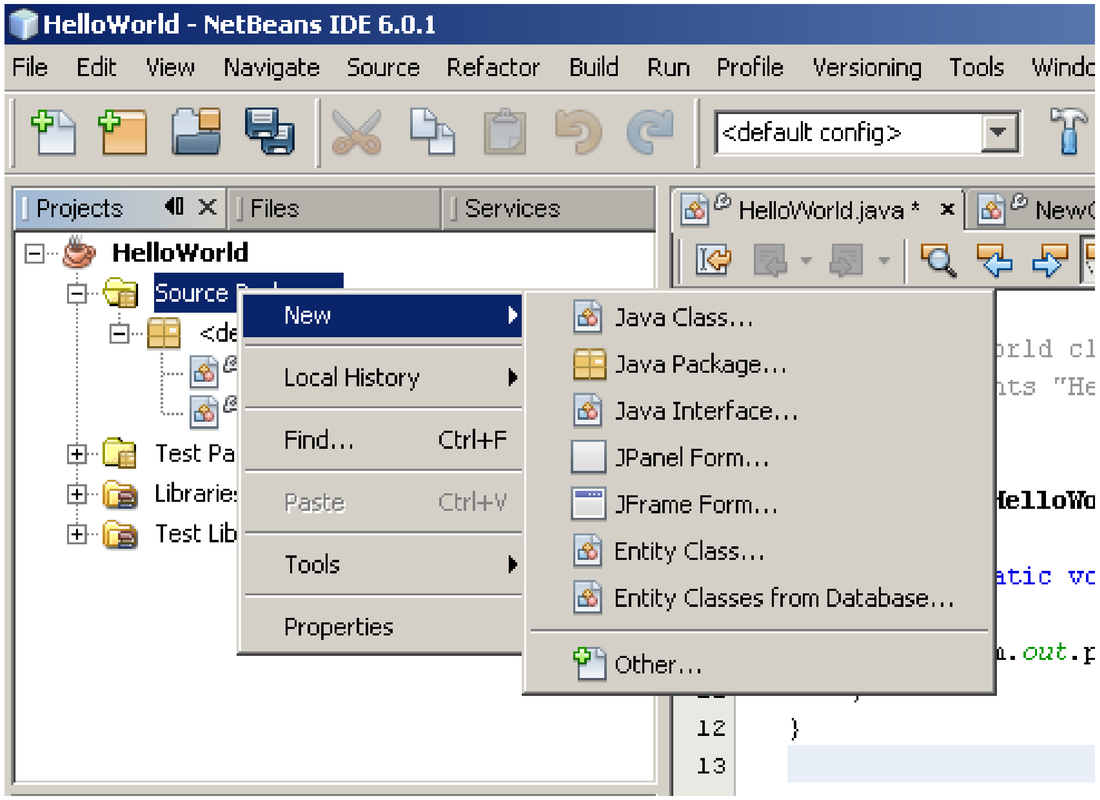


```
/**
 * The HelloWorld class implements an application that
 * simply prints "Hello World!" to standard output.
 */

public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello World!"); // Display the string.
    }
```

```

public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello World!"); // Display the string.
    }
}

```

Creating Files

- In the Projects window, right-click the Source Packages node and choose one of the templates from the New submenu.



Opening Files

- You can display a file in the Source Editor by double-clicking the file in the Projects or Files window.
- It should open in the Source Editor portion of the IDE.

Code Folding

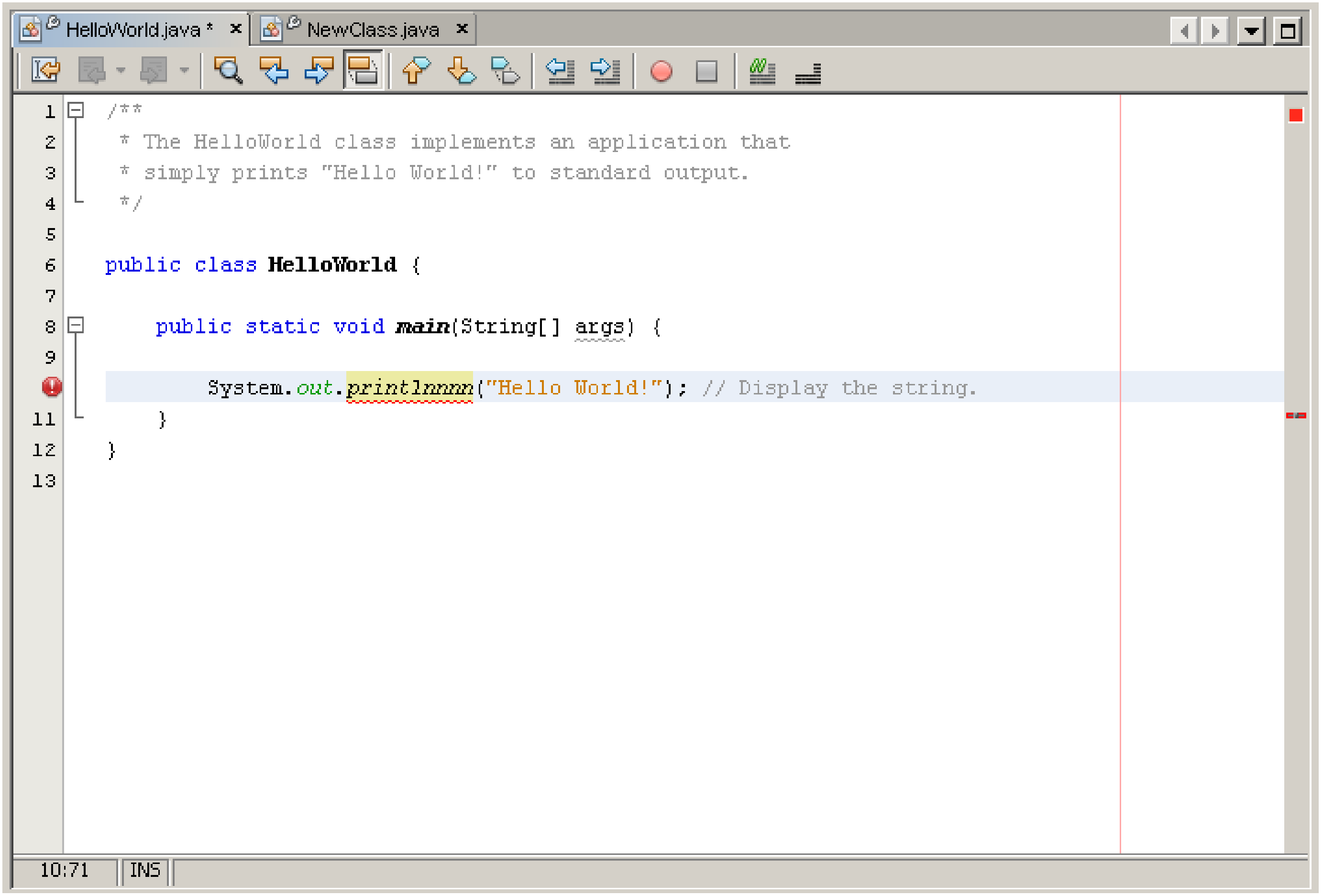
- For each section of comments and each method name, notice the **minus icon** and the **line extending** below it.
- This denotes a piece of text that can be folded, or hidden.
- Code folding can be enabled or disabled
  - To disable code folding in NetBeans, select Tools | Options | Editor

Current-Line Highlighting

- A useful feature of the NetBeans Source Editor is current-line highlighting.
- The line that contains the cursor is **lightly highlighted**, so you always know exactly which line is being edited.

Syntax-Error Highlighting

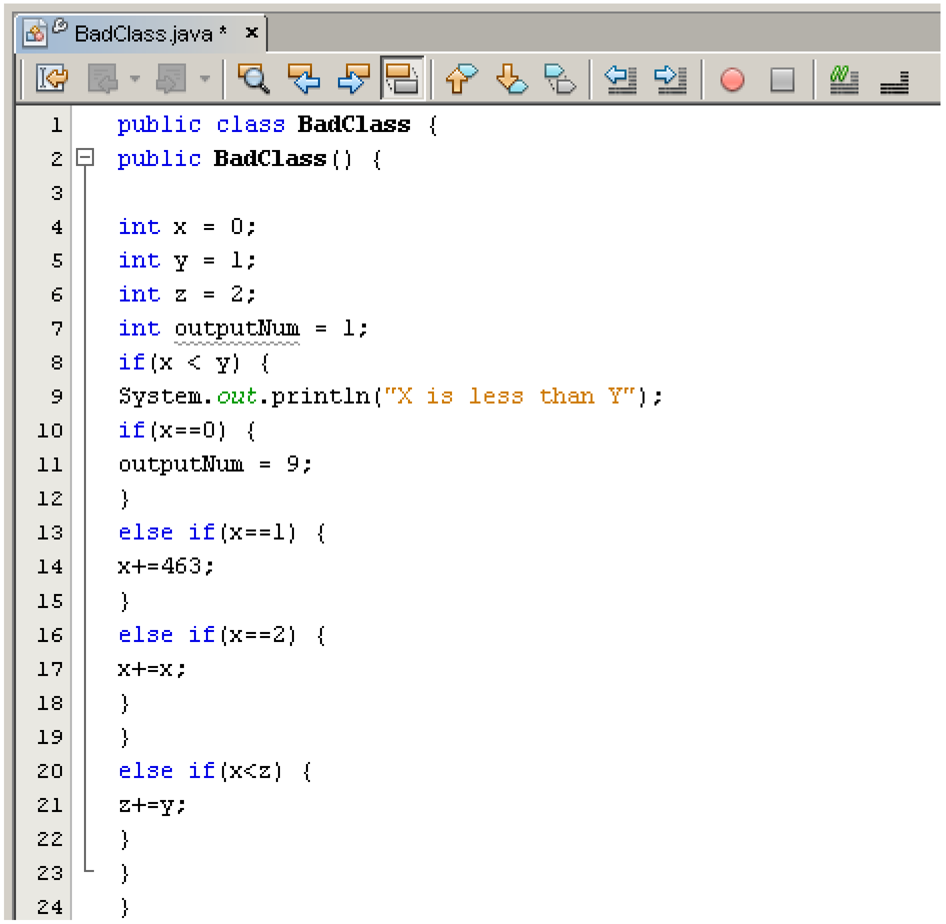
- The code syntax-error highlighting is another feature of NetBeans.



```
/**
 * The HelloWorld class implements an application that
 * simply prints "Hello World!" to standard output.
 */

public class HelloWorld {

    public static void main(String[] args) {

        System.out.printlnnnn("Hello World!"); // Display the string.
    }
}
```

Code Indentation

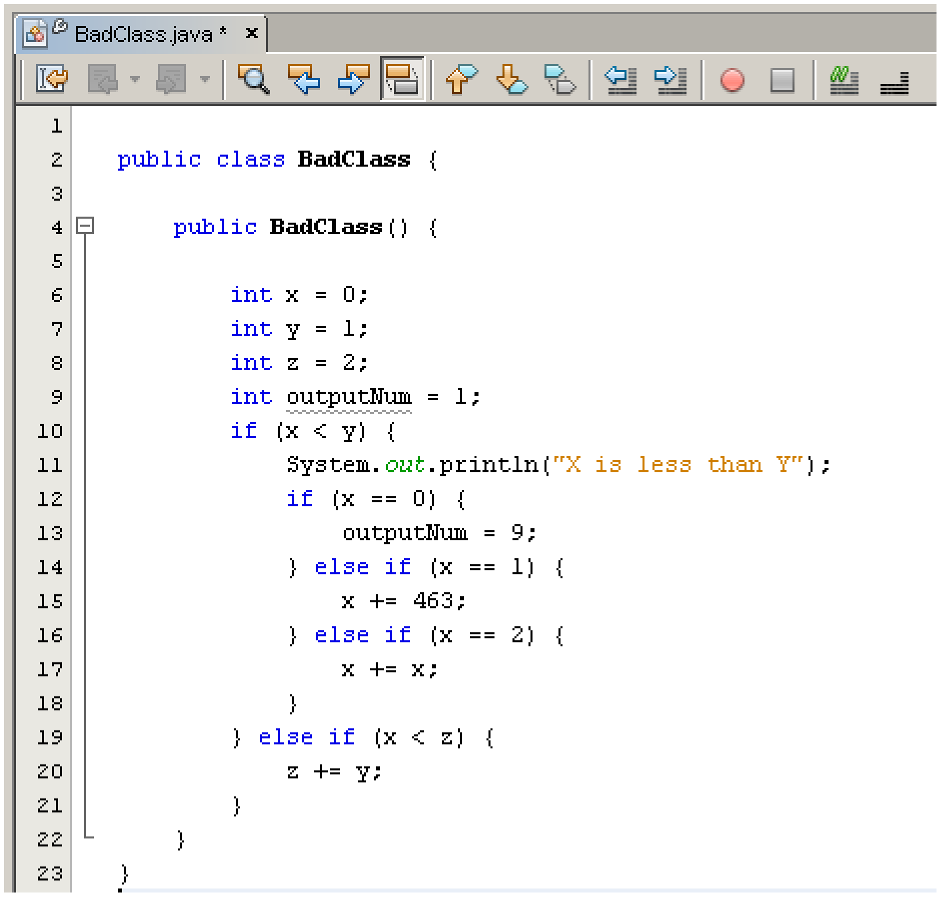
- Formatting your code and indenting each line properly makes the code more readable and easier to maintain.



```java
public class BadClass {
public BadClass() {

int x = 0;
int y = 1;
int z = 2;
int outputNum = 1;
if(x < y) {
System.out.println("X is less than Y");
if(x==0) {
outputNum = 9;
}
else if(x==1) {
x+=463;
}
else if(x==2) {
x+=x;
}
}
else if(x<z) {
z+=y;
}
}
}
```

Code Indentation

- With the Source | Format option, you can create code and have NetBeans enforce good indentation.



```

public class BadClass {

    public BadClass() {

        int x = 0;
        int y = 1;
        int z = 2;
        int outputNum = 1;
        if (x < y) {
            System.out.println("X is less than Y");
            if (x == 0) {
                outputNum = 9;
            } else if (x == 1) {
                x += 463;
            } else if (x == 2) {
                x += x;
            }
        } else if (x < z) {
            z += y;
        }
    }
}
```

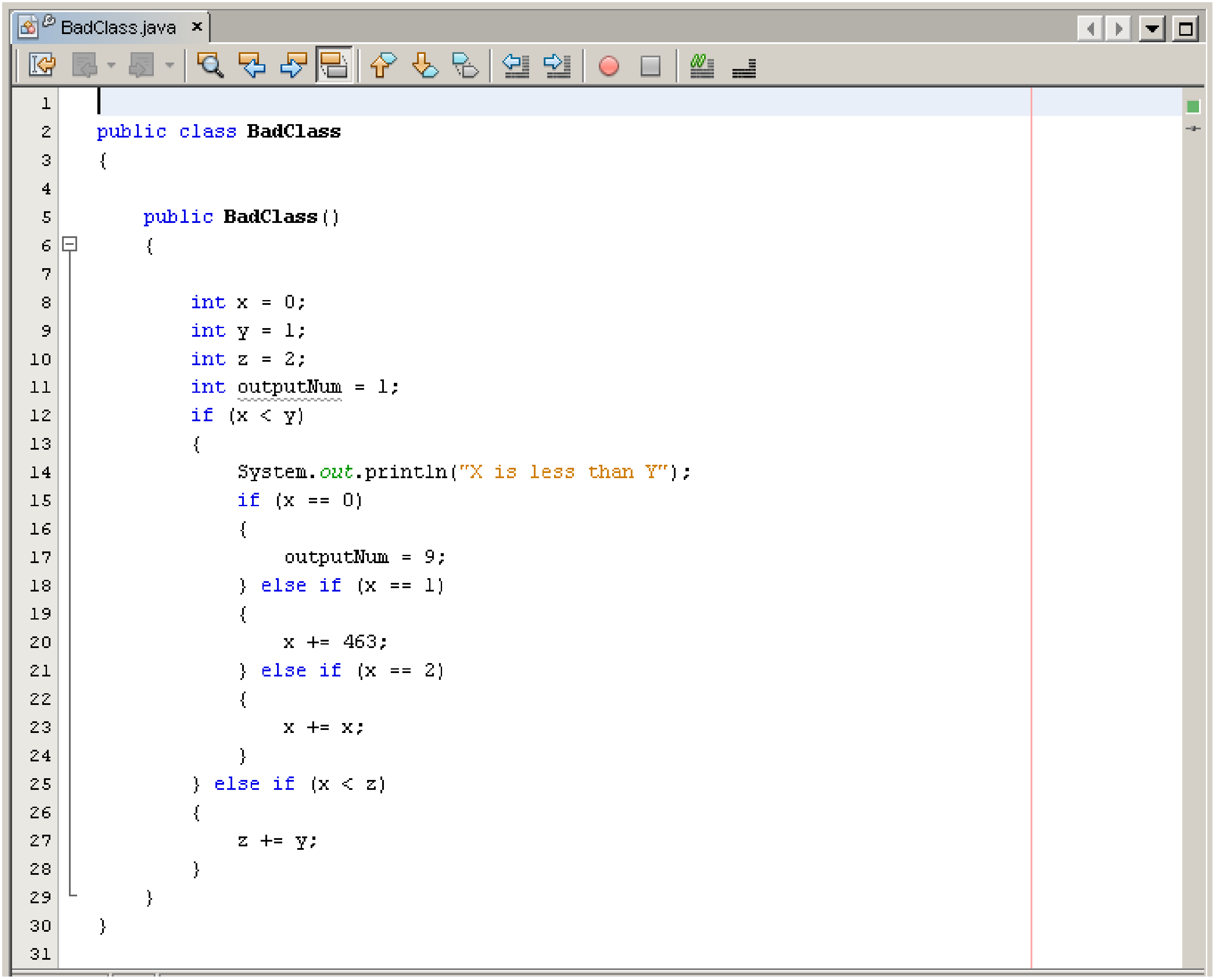
Setting Code Editor Indentation

- NetBeans allows some flexibility when configuring code indentation and formatting
  - select Source | Format Code, for formatting source code
- You can modify code styles by choosing
  - Tools | Options | Editor and clicking the Indentation tab.

Setting Braces Placement

- You can modify braces placement by choosing
  - Tools | Options | Java Code and clicking the Formatting tab and selecting Alignment and Braces option
- We use following setting:
  - Class Declaration: New Line
  - Class Declaration: New Line
  - Other: New Line

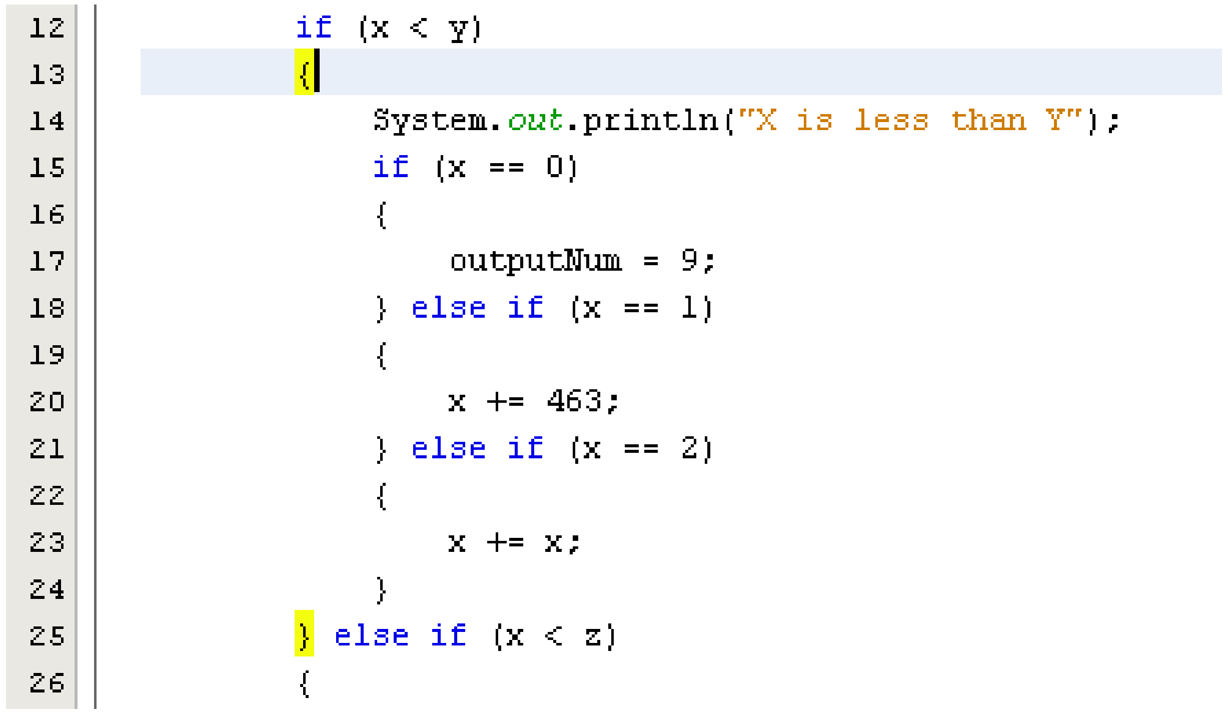
Setting Braces Placement


BadClass.java

```

public class BadClass
{

    public BadClass()
    {

        int x = 0;
        int y = 1;
        int z = 2;
        int outputNum = 1;
        if (x < y)
        {
            System.out.println("X is less than Y");
            if (x == 0)
            {
                outputNum = 9;
            } else if (x == 1)
            {
                x += 463;
            } else if (x == 2)
            {
                x += x;
            }
        } else if (x < z)
        {
            z += y;
        }
    }
}

```

Identifying Starting and Ending Braces

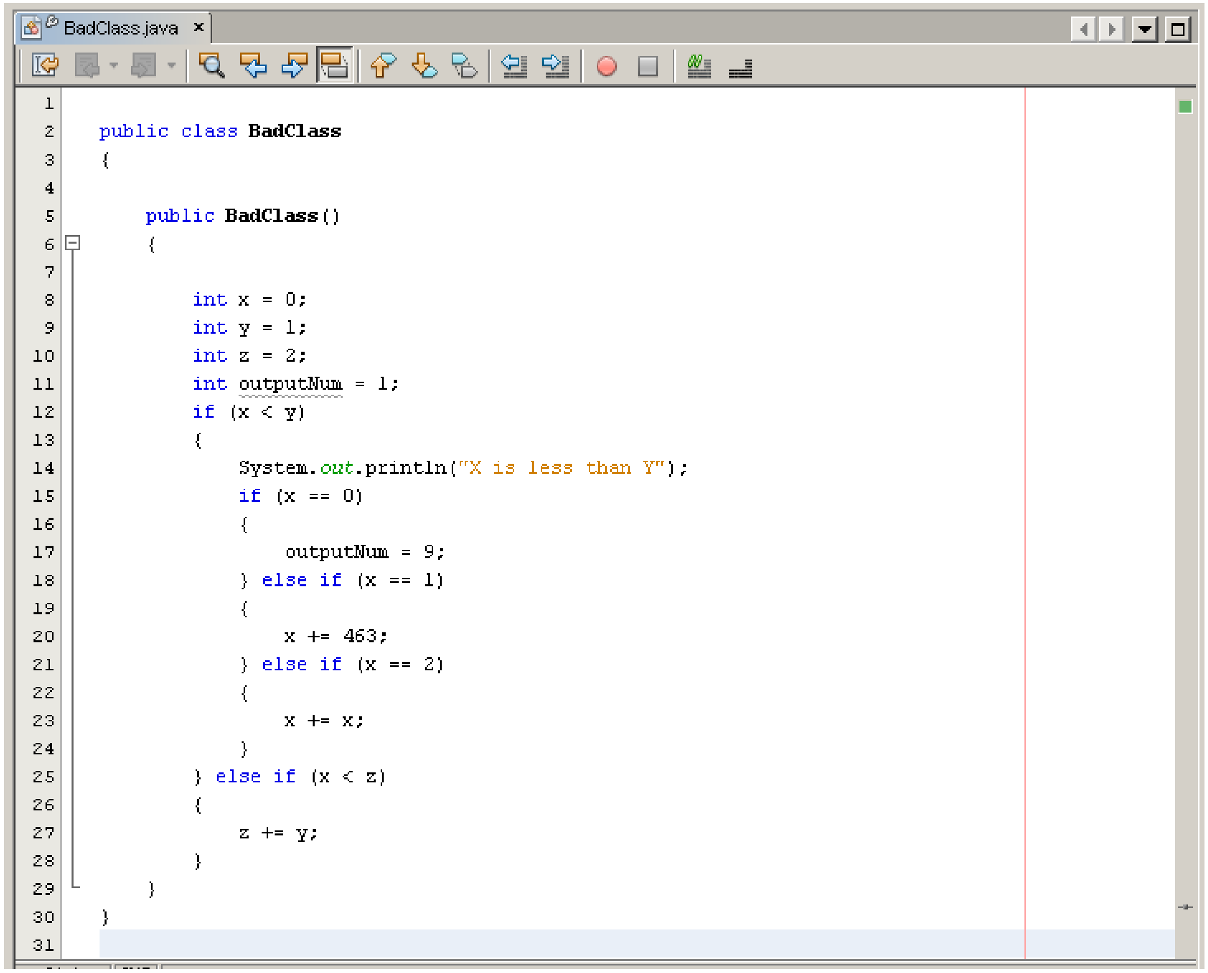
- If you click next to the curly brace at the end of line 13, then that curly brace should be highlighted



```
12          if (x < y)
13          {
14              System.out.println("X is less than Y");
15              if (x == 0)
16              {
17                  outputNum = 9;
18              } else if (x == 1)
19              {
20                  x += 463;
21              } else if (x == 2)
22              {
23                  x += x;
24              }
25          } else if (x < z)
26          {
```

Identifying Unused Variables

- Another feature of NetBeans 6 is the ability to see all unused variables. ouptputNum in this program:



```

public class BadClass
{

    public BadClass()
    {

        int x = 0;
        int y = 1;
        int z = 2;
        int outputNum = 1;
        if (x < y)
        {
            System.out.println("X is less than Y");
            if (x == 0)
            {
                outputNum = 9;
            } else if (x == 1)
            {
                x += 463;
            } else if (x == 2)
            {
                x += x;
            }
        } else if (x < z)
        {
            z += y;
        }
    }
}
```

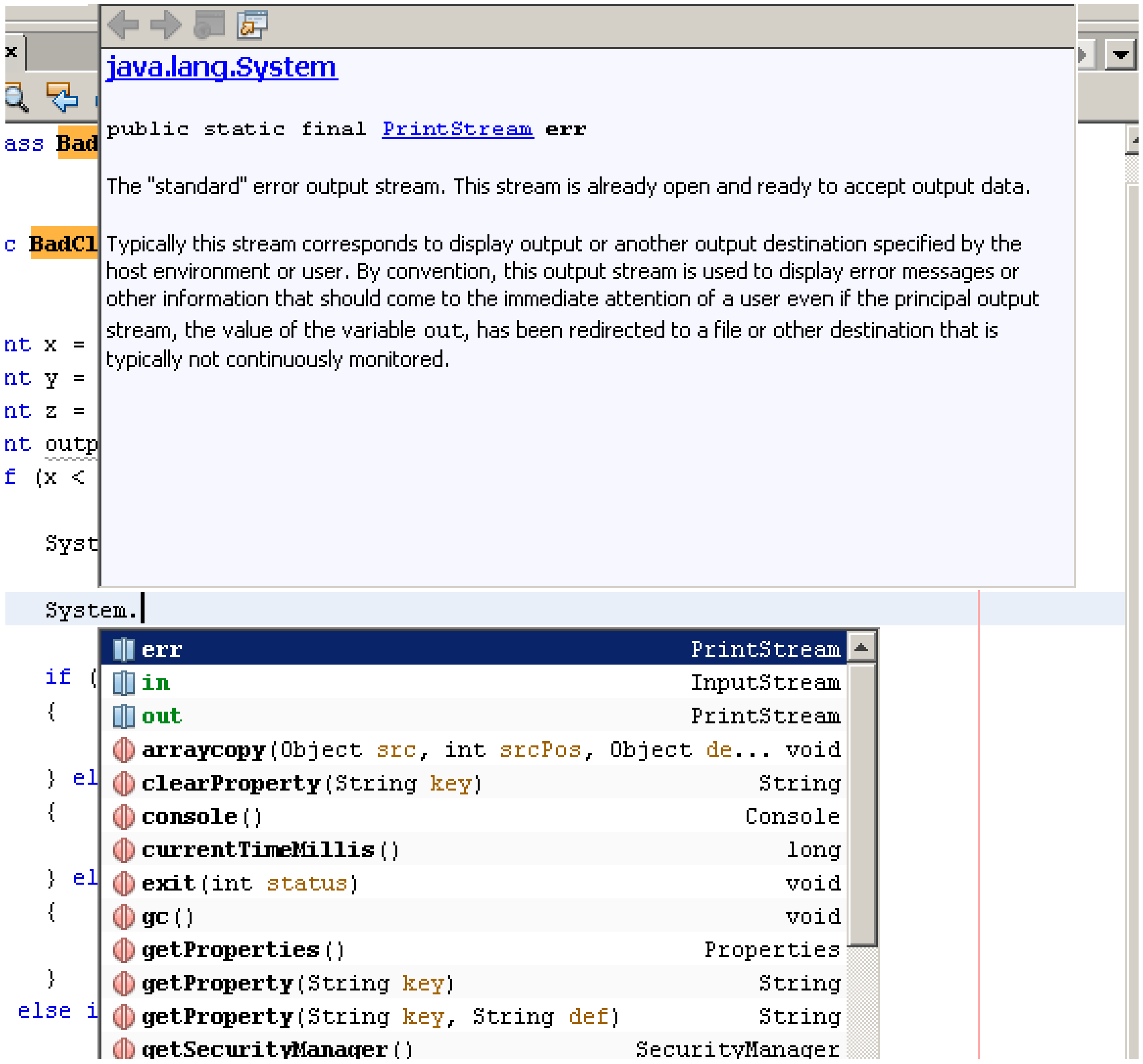
Code Completion

- Code completion allows you to enter the name of a class, interface, package, field, or method without having to type the entire name.
- Keystrokes affecting code completion

Keystroke	Action
Ctrl+Space	Force the code completion pop-up to appear.
Enter	Insert the selected item into your code.
Escape	Close the code completion box and cancel any text insertions.
Up arrow	Scroll through list of items.
Down arrow	Scroll through list of items.
Page-Up	Scroll to top of visible list of items.
Page-Down	Scroll to bottom of visible list of items.
Home	Scroll to absolute top of the entire list of items.
End	Scroll to absolute bottom of the entire list of items.

NetBeans IDE 6.0

Code Completion



java.lang.System

```
public static final PrintStream err
```

The "standard" error output stream. This stream is already open and ready to accept output data.

Typically this stream corresponds to display output or another output destination specified by the host environment or user. By convention, this output stream is used to display error messages or other information that should come to the immediate attention of a user even if the principal output stream, the value of the variable out, has been redirected to a file or other destination that is typically not continuously monitored.

```
ass Bad

c BadCl

nt x =
nt y =
nt z =
nt outp
f (x <

   Syst

   System.

   if (
   {

   } el
   {

   } el
   {

   }
  else i
```

```
err                                               PrintStream
in                                                InputStream
out                                               PrintStream
arraycopy(Object src, int srcPos, Object de... void
clearProperty(String key)                         String
console()                                         Console
currentTimeMillis()                               long
exit(int status)                                  void
gc()                                              void
getProperties()                                   Properties
getProperty(String key)                           String
getProperty(String key, String def)               String
getSecurityManager()                              SecurityManager
```

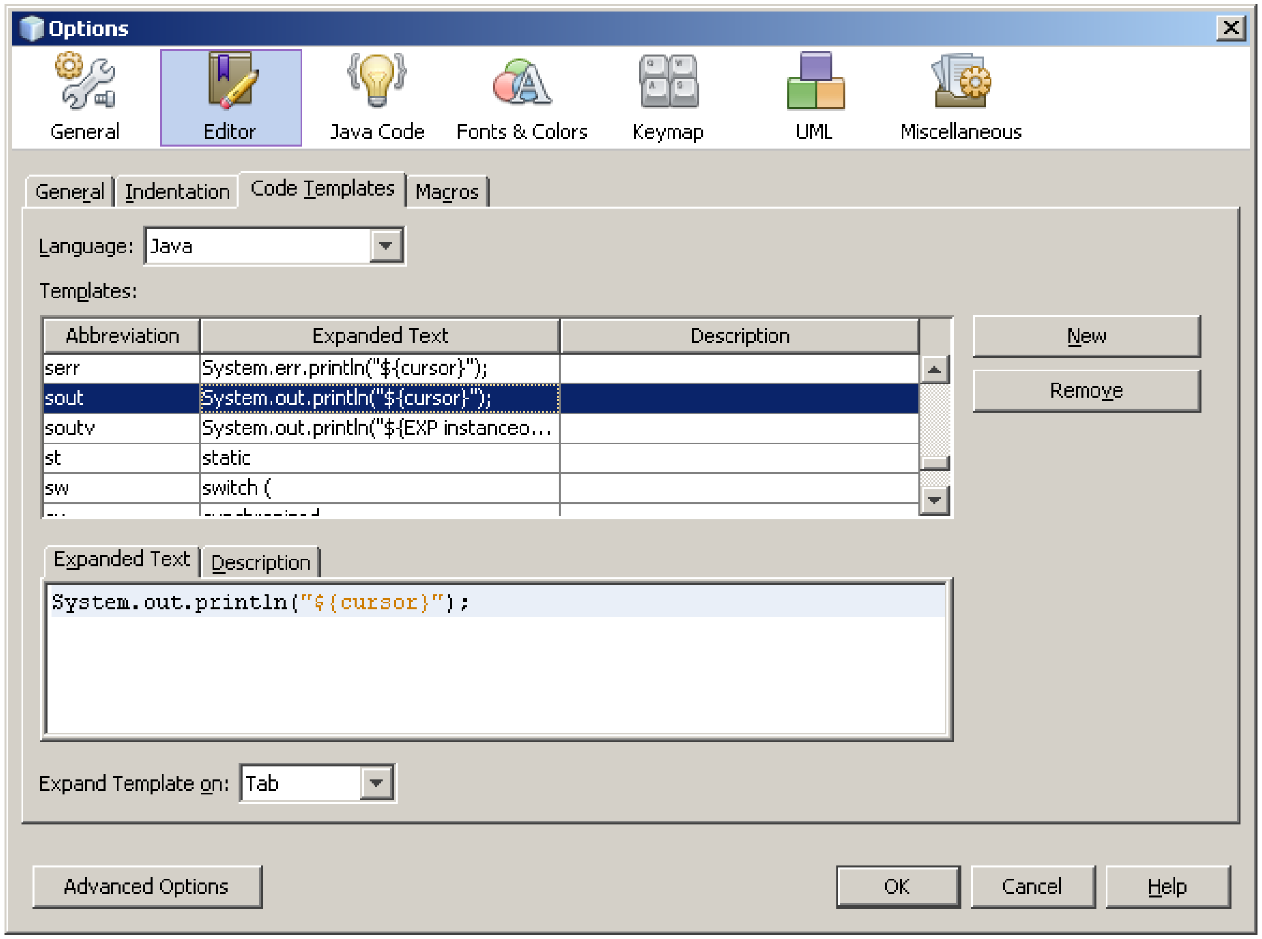
Code Templates

- Code templates allow you to insert a block of code or text automatically by typing a few characters.
- At first, you have to remember the correct abbreviation for the code template you want.
- Examples: you can simply type sout and press the Tab. The sout text is expanded into this:

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

Customizing Templates

- To view the list of code templates: select Tools | Options | Editor and click the Code Templates tab.



Language: Java

Templates:

Abbreviation	Expanded Text	Description
serr	System.err.println("${cursor}");	
sout	System.out.println("${cursor}");	
soutv	System.out.println("${EXP instanceo...	
st	static	
sw	switch (	

Expanded Text | Description

```
System.out.println("${cursor}");
```

Expand Template on: Tab

Configuring Keymaps

- Every good software tool should provide shortcut keys (also known as hotkeys).
- Many menu commands, actions, and tools can be activated via keyboard shortcuts.
- NetBeans categorizes a group of shortcuts as a keymap*.*
- Keymaps can be configured in the Basic Options window.
  - Select Tools | Options and choose Keymap.

Customizing the IDE

Line Numbers

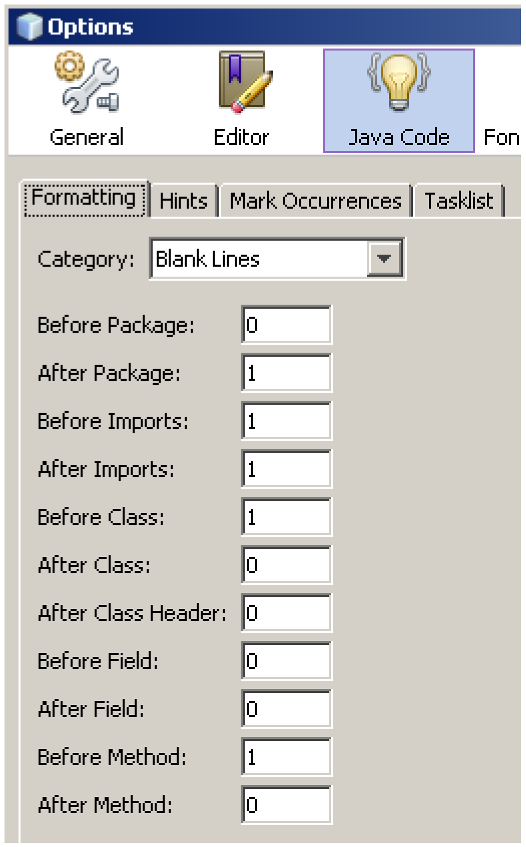
- In the Source Editor, line numbers are displayed along the left column.
  - To track where certain pieces of code
  - A quick way to trace the location of exceptions that are thrown.
- If the line numbers are not displayed, enable them by selecting View | Show Line Numbers.

Setting Braces Placement

- You can modify braces placement by choosing
  - Tools | Options | Java Code and clicking the Formatting tab and selecting Alignment and Braces option
- We use following setting:
  - Class Declaration: New Line
  - Method Declaration: New Line
  - Other: New Line

Setting Blank Lines

- You can modify braces placement by choosing
  - Tools | Options | Java Code and clicking the Formatting tab and selecting Blank Lines option
- We use this setting:

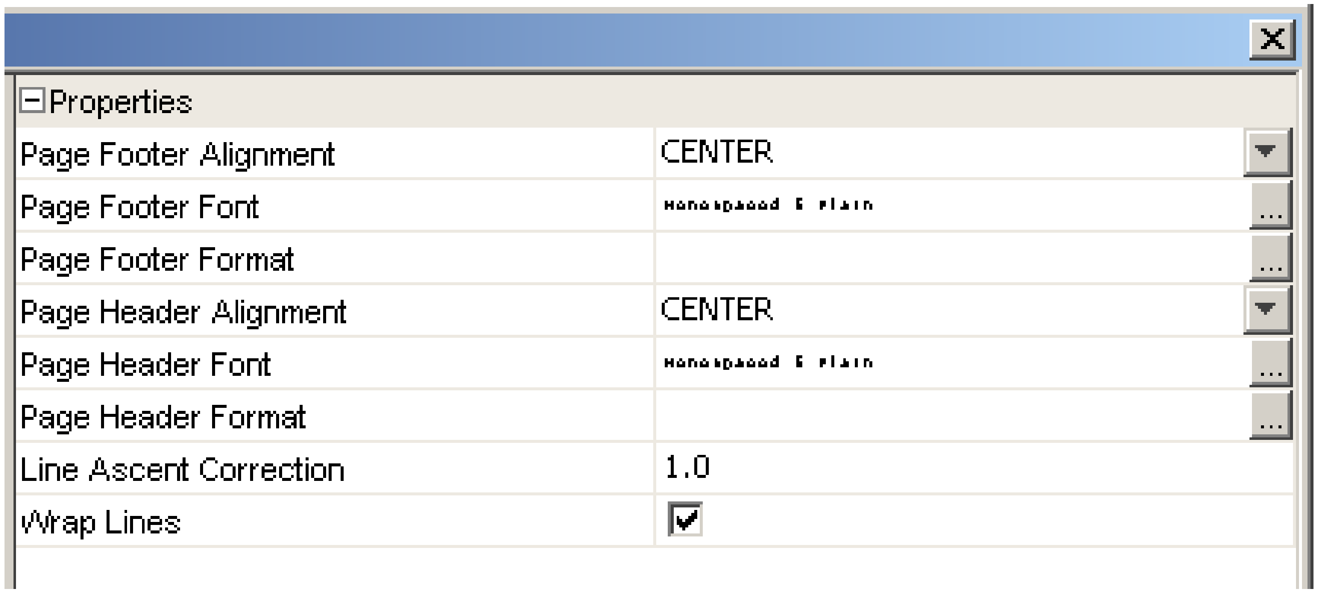


Category: Blank Lines

Before Package:	0
After Package:	1
Before Imports:	1
After Imports:	1
Before Class:	1
After Class:	0
After Class Header:	0
Before Field:	0
After Field:	0
Before Method:	1
After Method:	0

Print Setting

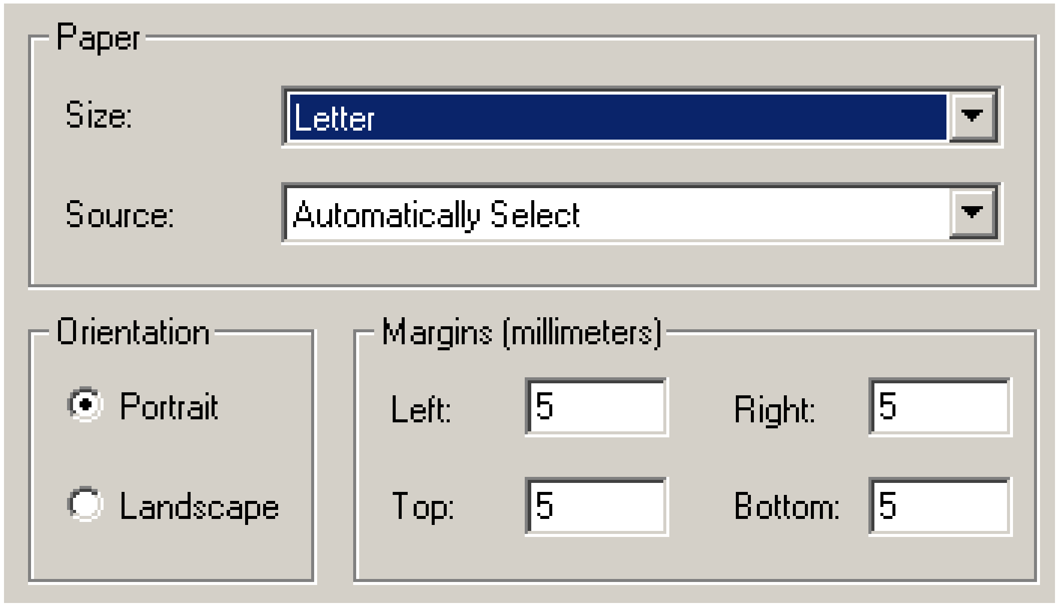
- You can modify print setting by choosing
  - Tools | Options | Advance Options and clicking the System and selecting Print Setting
- We use:



Page Footer Alignment	CENTER
Page Footer Font	Monospaced 6 Plain
Page Footer Format	
Page Header Alignment	CENTER
Page Header Font	Monospaced 6 Plain
Page Header Format	
Line Ascent Correction	1.0
Wrap Lines	✓

Print Setting

- You can modify page setup for printing by choosing:
  - File | Page Setup
- We use:



Paper

Size: Letter

Source: Automatically Select

Orientation

[✓ Portrait] Landscape

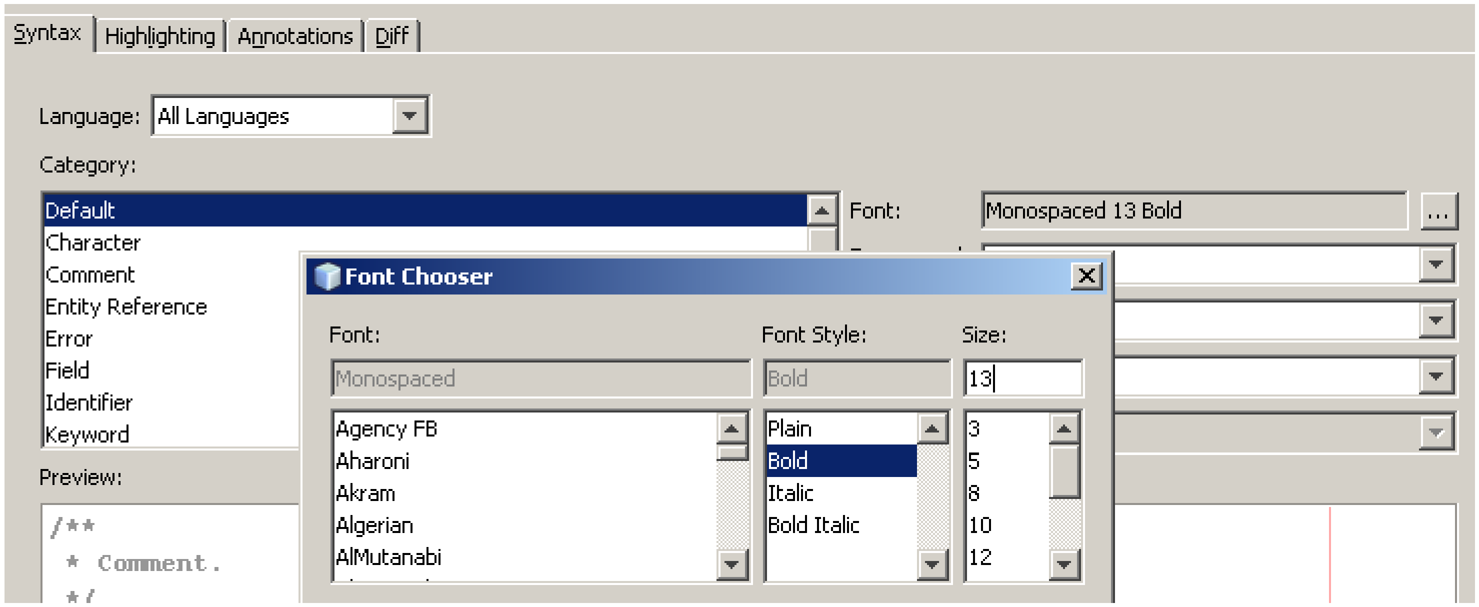
Margins (millimeters)

Left: 5 Right: 5

Top: 5 Bottom: 5

Choosing Fonts and Colors

- The font and color customizations are grouped into a color profile.
- To customize color profiles
  - select Tools | Options | Fonts & Colors and select the Syntax tab.
- We use:



Language: All Languages

Category:

Default
Character
Comment
Entity Reference
Error
Field
Identifier
Keyword

Font: Monospaced 13 Bold

Preview:

```
/**
 * Comment.
 */
```

Font Chooser

Font: Monospaced

Font Style: Bold

Size: 13

Agency FB
Aharoni
Akram
Algerian
AlMutanabi

Plain
Bold
Italic
Bold Italic

3
5
8
10
12

References

References

- Patrick Keegan, et. al., **NetBeans™ IDE Field Guide: Developing Desktop, Web, Enterprise, and Mobile Applications**, Second Edition, Prentice Hall, 2006. (Chapter 1 & 2)

- Adam Myatt, **Pro NetBeans™ IDE 6, Rich Client Platform Edition**, Springer-Verlag New York, 2008. (Chapter 1 & 2)

The End