Event-Driven Programs

CS 1025 Computer Science Fundamentals I

Stephen M. Watt

University of Western Ontario

Command Line vs Events

- So far, all our programs have started from a “main” method which calls other methods, takes input, makes objects, prints output, etc.
- Another model of programming is to have programs that respond to “events.”
- These events could be
  - Mouse clicks
  - Key presses
  - Pressing a brake pedal
  - Lifting a telephone receiver

Event-Driven Code

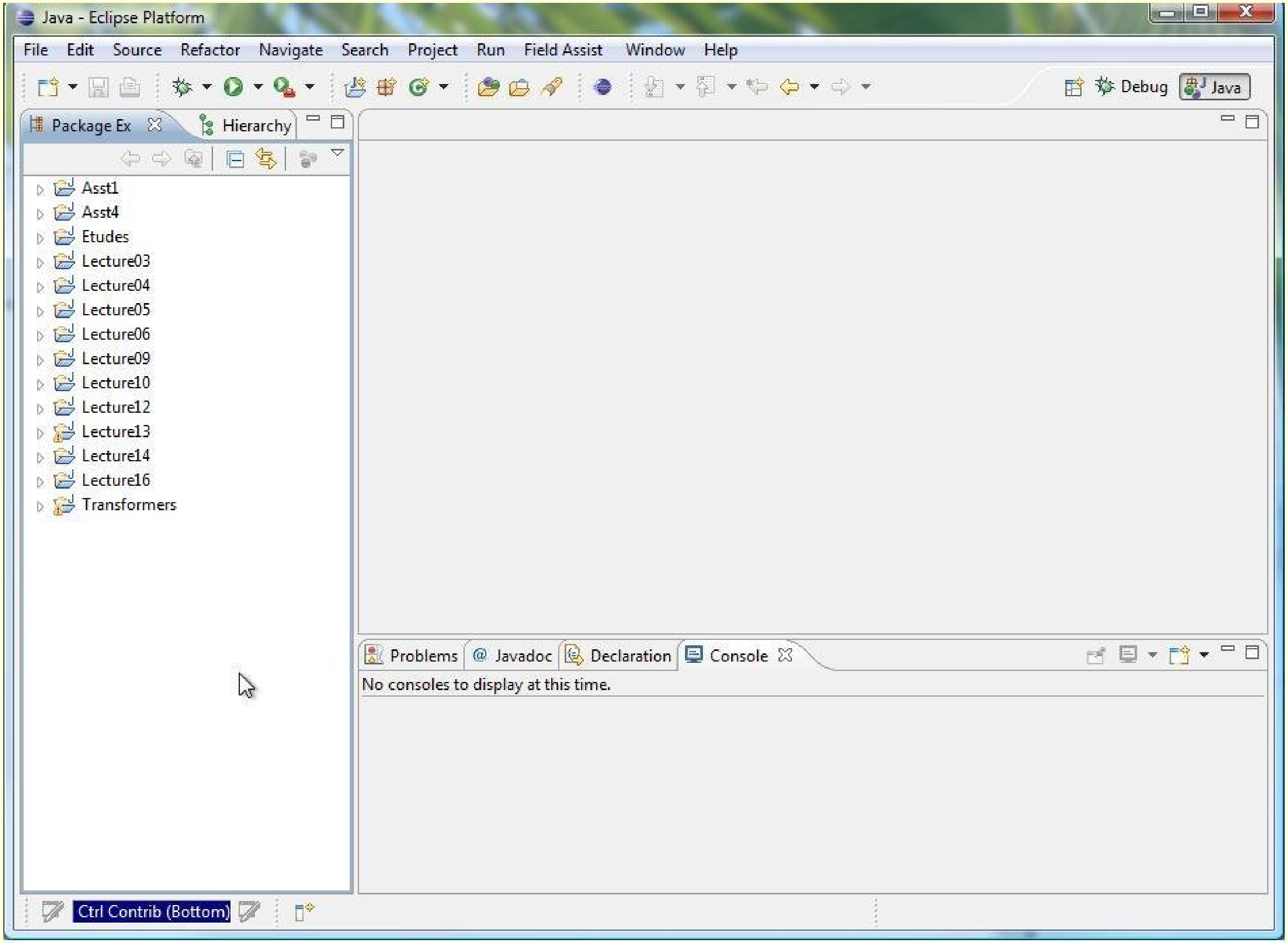
- Event-driven code associates particular functions or methods to be called when specified events occur.
- In Java, this is done by creating “handler” objects from which specific methods get invoked.
- These handler objects are then placed on the things that receive events, like buttons, scroll bars etc.

GUIs in Java

- Graphical user interfaces are programmed in Java using
  - the abstract window toolkit (old, lower level) and
  - Swing (new, higher level)

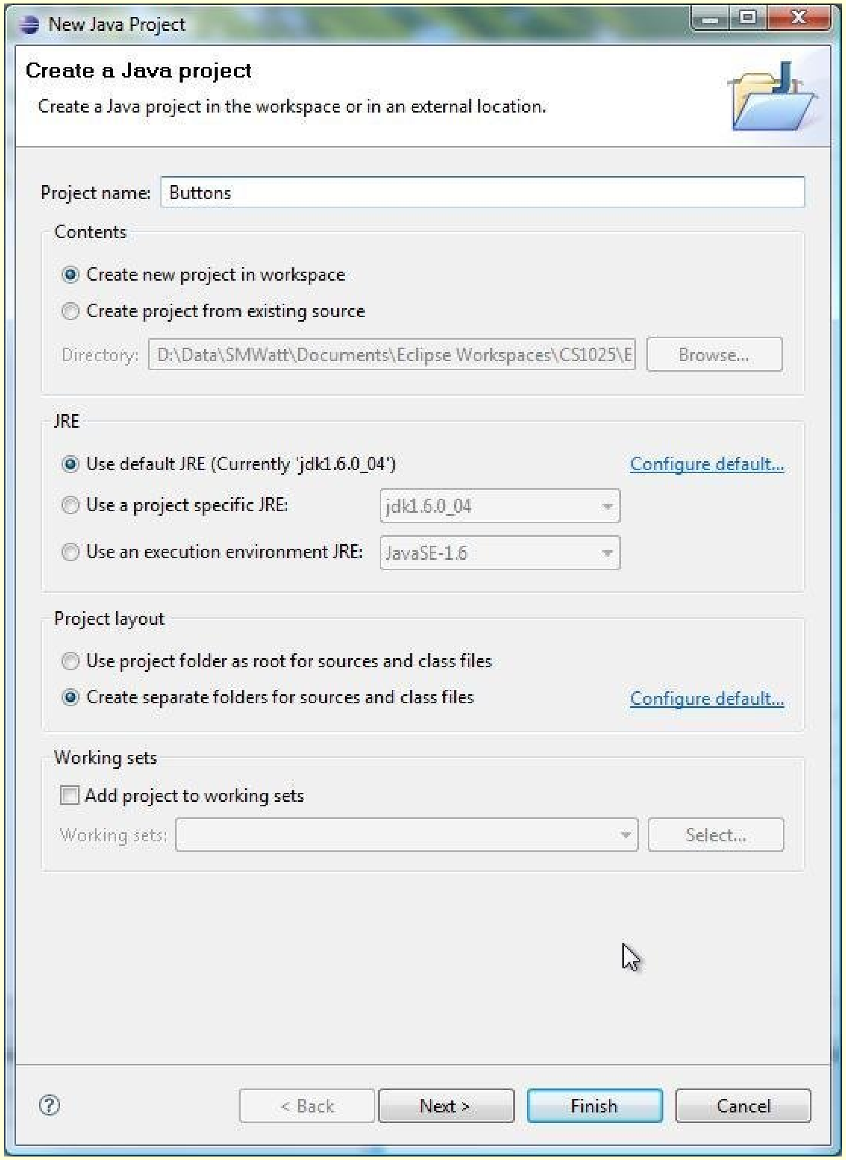
- The next slides give an example of creating a simple Swing application.

Start Eclipse



No consoles to display at this time.

Create a Project



New Java Project

Create a Java project

Create a Java project in the workspace or in an external location.

Project name: Buttons

Contents

● Create new project in workspace

○ Create project from existing source

Directory: D:\Data\SMWatt\Documents\Eclipse Workspaces\CS1025\E

JRE

● Use default JRE (Currently 'jdk1.6.0_04')

○ Use a project specific JRE: jdk1.6.0_04

○ Use an execution environment JRE: JavaSE-1.6

Project layout

○ Use project folder as root for sources and class files

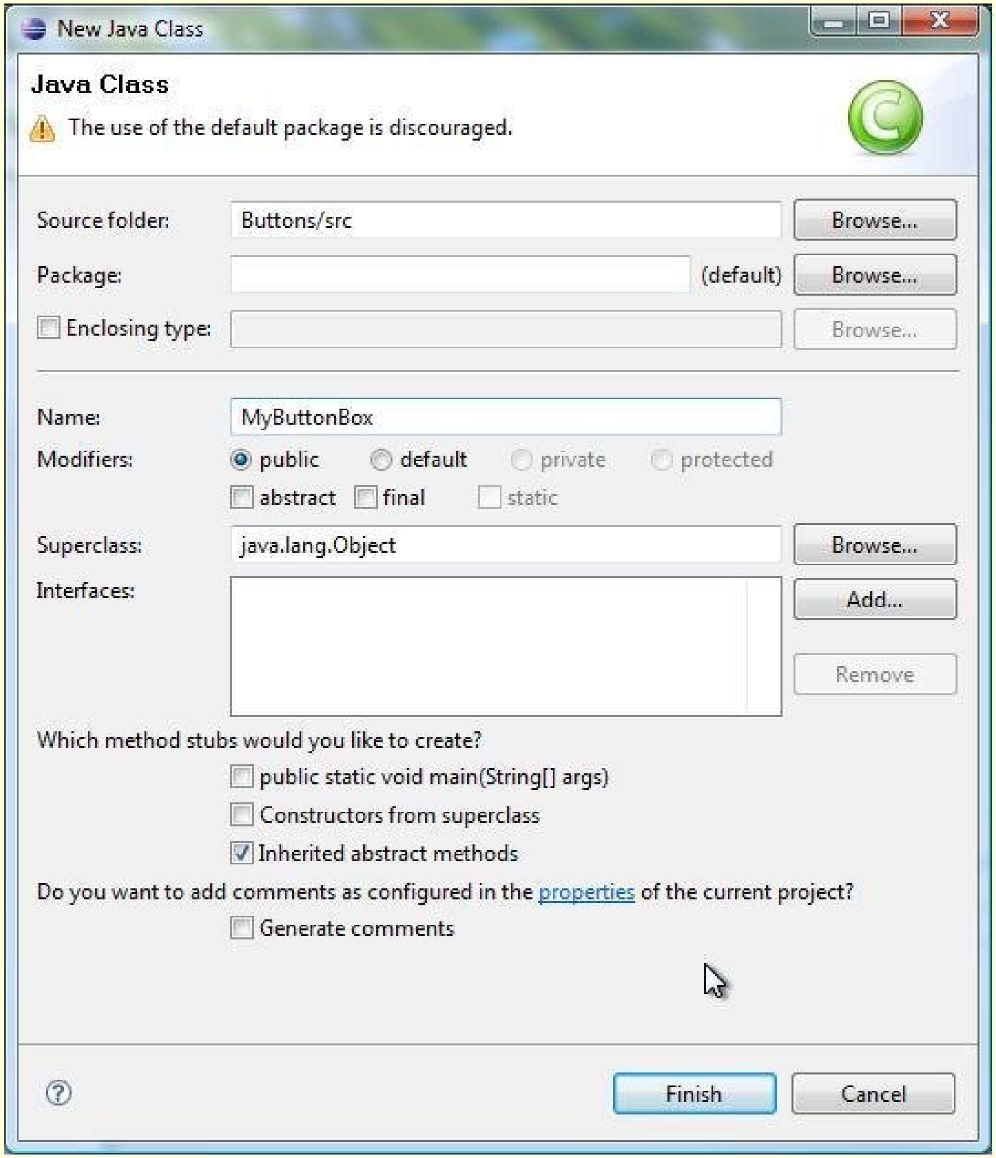
● Create separate folders for sources and class files

Working sets

□ Add project to working sets

Working sets:

Create a Class



Java Class

⚠ The use of the default package is discouraged.

Source folder: Buttons/src [Browse...]

Package: (default) [Browse...]

□ Enclosing type: [Browse...]

Name: MyButtonBox

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object [Browse...]

Interfaces: [Add...] [Remove]

Which method stubs would you like to create?

□ public static void main(String[] args)

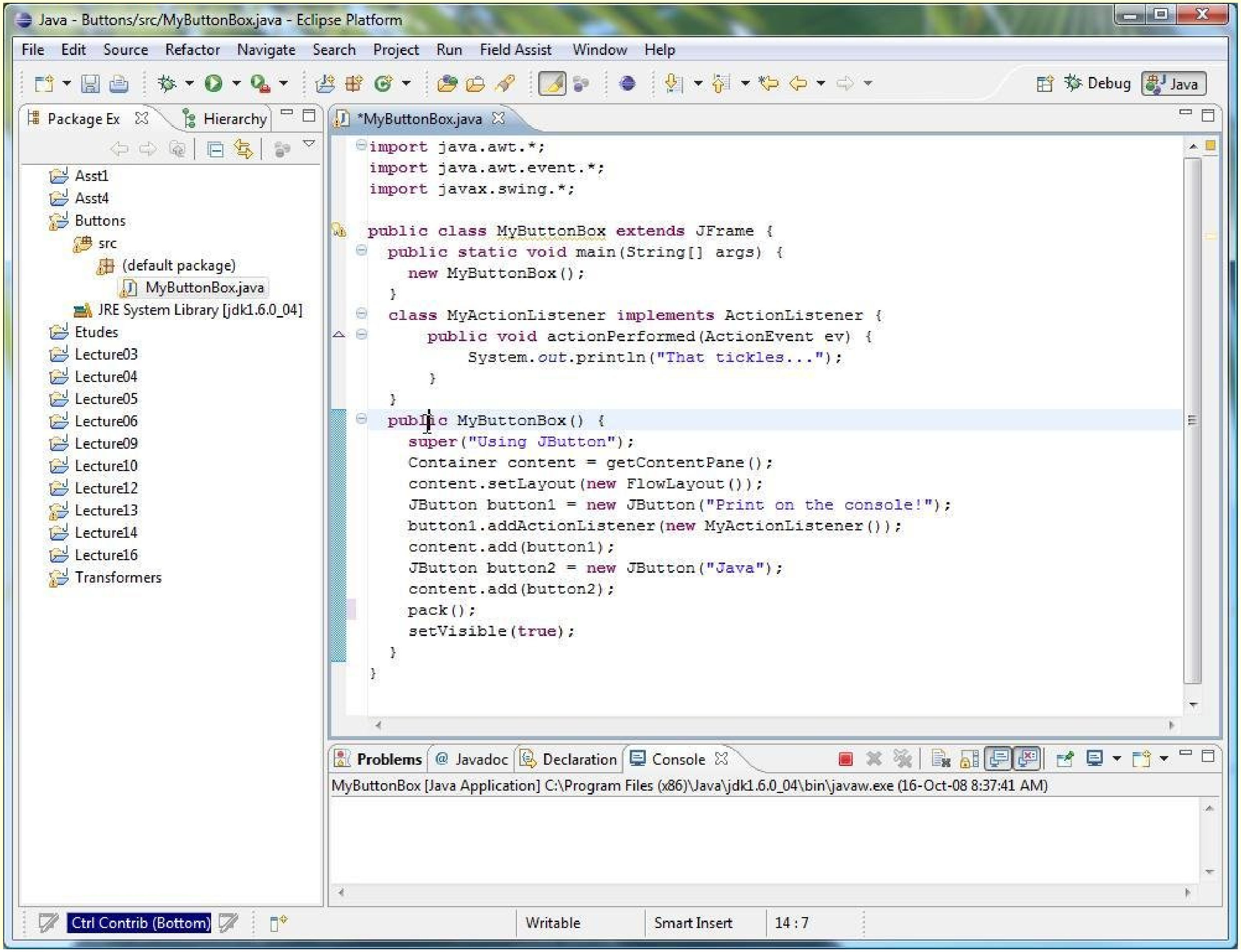
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

□ Generate comments

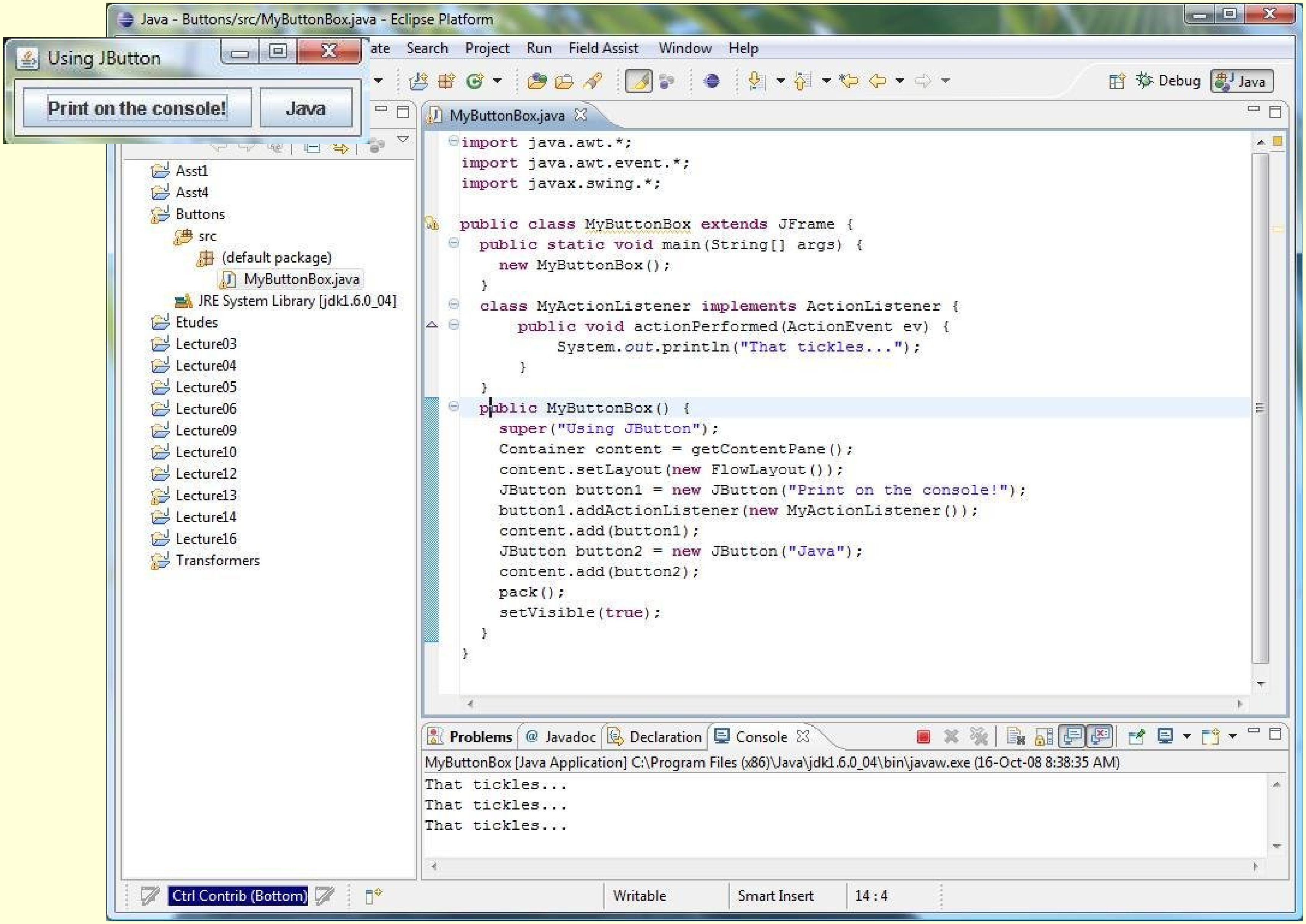
The Program – note the Listener



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyButtonBox extends JFrame {
  public static void main(String[] args) {
    new MyButtonBox();
  }
  class MyActionListener implements ActionListener {
      public void actionPerformed(ActionEvent ev) {
          System.out.println("That tickles...");
      }
  }
  public MyButtonBox() {
    super("Using JButton");
    Container content = getContentPane();
    content.setLayout(new FlowLayout());
    JButton button1 = new JButton("Print on the console!");
    button1.addActionListener(new MyActionListener());
    content.add(button1);
    JButton button2 = new JButton("Java");
    content.add(button2);
    pack();
    setVisible(true);
  }
}
```

```
MyButtonBox [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_04\bin\javaw.exe (16-Oct-08 8:37:41 AM)
```



```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyButtonBox extends JFrame {
  public static void main(String[] args) {
    new MyButtonBox();
  }
  class MyActionListener implements ActionListener {
      public void actionPerformed(ActionEvent ev) {
          System.out.println("That tickles...");
      }
  }
  public MyButtonBox() {
    super("Using JButton");
    Container content = getContentPane();
    content.setLayout(new FlowLayout());
    JButton button1 = new JButton("Print on the console!");
    button1.addActionListener(new MyActionListener());
    content.add(button1);
    JButton button2 = new JButton("Java");
    content.add(button2);
    pack();
    setVisible(true);
  }
}
```

MyButtonBox [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_04\bin\javaw.exe (16-Oct-08 8:38:35 AM)

```
That tickles...
That tickles...
That tickles...
```