Java Programming
Unit 6

Inner Classes. Intro to HTML and Applets. Installing Apache Tomcat Server.

Swing Adapters

Swing adapters are classes that implement empty functions required by listener interfaces. You need to override only the methods you are interested in.

```
class MyWindowEventProcessor extends java.awt.WindowsAdapter {

  public void windowClosing(WindowEvent e) {
    // your code goes here.
    // For example, ask if the user wants to save data
  }
}
```

The code above is more compact than writing class `MyWindowEventProcessor` that implements `WindowListener` with all the methods below:

```
windowActivated (WindowEvent)
windowClosed(WindowEvent)
windowClosing(WindowEvent)
windowDeactivated (WindowEvent)
windowDeiconified(WindowEvent)
windowIconified(WindowEvent)
windowOpened(WindowEvent)
```

Let Adapter be a Listener

Imagine, that the `Calculator` class includes the following code:

```
MyWindowEventProcessor mw = new MyWindowEventProcessor();

this.addWindowListener(mw);
```

All window's events will be processed by the methods of the object `MyWindowEventProcessor`. In this case we have a named class that processes events.

Do we really need to create a separate class just for processing one event in our window?

Inner Classes

```
class Tax{
    double grossIncome;
    int dependents;

   double calcStateTax(){

        TaxOptimizer tOpt = new  TaxOptimizer();
        tOpt.optimize(grossIncome, dependents);
   }

      public TaxOptimizer getTaxOptimizer(){

          return new TaxOptimizer();
      }

     class TaxOptimizer {

       int taxCode;

       void setTaxCode(int tCode){
              taxCode=tCode;
       }

       int optimize(double grossIncome, int dep){
          // Some optimization code goes here
       }
 }
```

A class defined inside another one is called an *inner class*.

The method `getTaxOptimizer()` returns an instance of the inner class.

If a class `TestTax` need to call the method `setTaxCode()` from
the inner class it can:

```
Tax t = new Tax(2, "NY", 50000);
Tax.TaxOptimizer tOptimizer = t.getTaxOptimizer();
tOptimizer.setTaxCode(12345);
```

Here's another way to produce the same result:

```
Tax t = new Tax(2, "NY", 50000);
Tax.TaxOptimizer tOptimizer =
                 t.new TaxOptimizer();
tOptimizer.setTaxCode(12345);
```

Anonymous Inner Classes

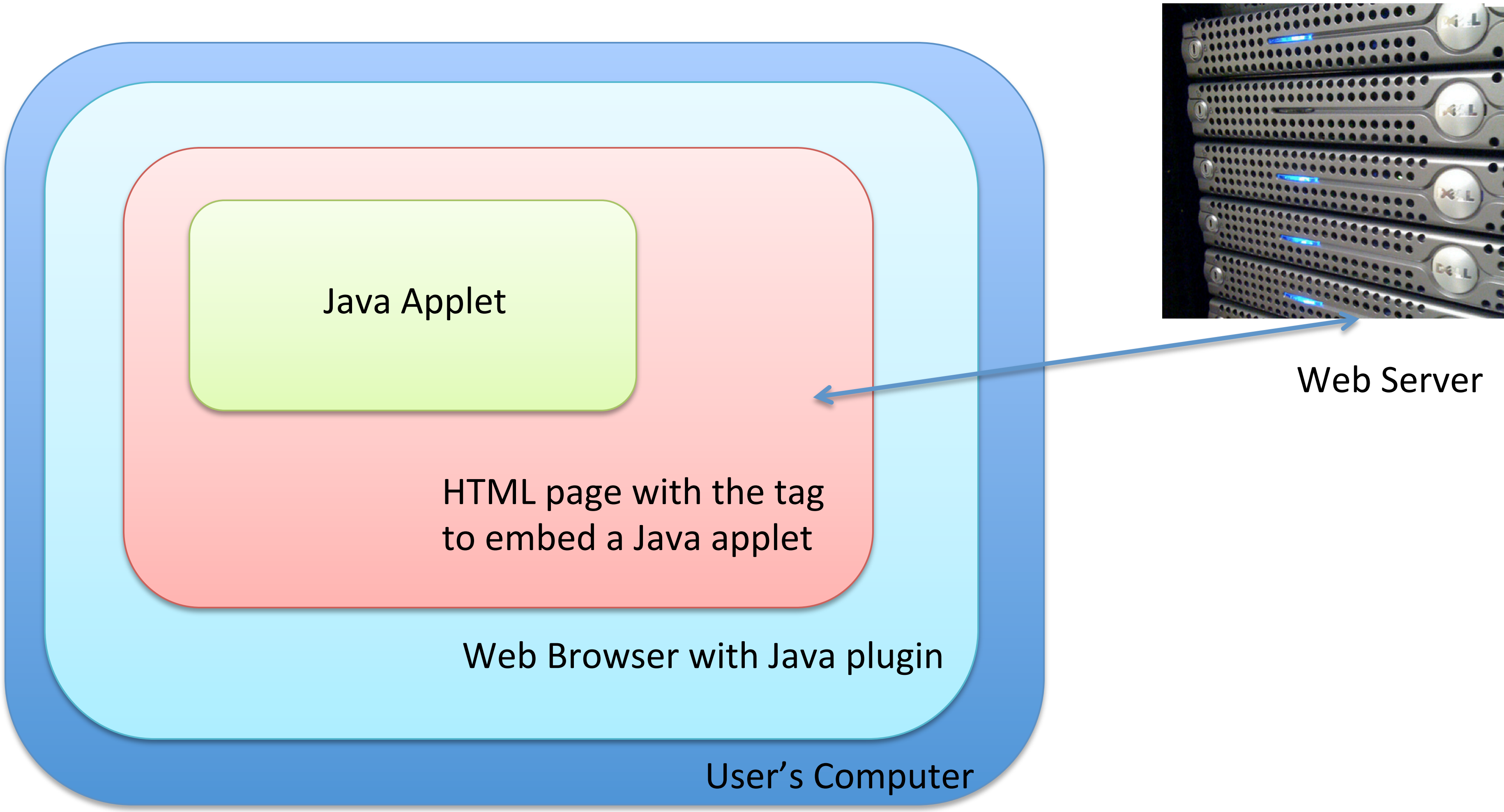
If an inner class does not have a name, it's called *anonymous*.

Imagine, that the `Calculator` class has the following code:

```
this.addWindowListener(
       new WindowAdapter() {
         public void windowClosing(WindowEvent e) {
              System.exit(0);
         }
        }
    );
```

Instead of providing a pre-created object (e.g. the `MyWindowEventProcessor` object) as an argument of the method `addWindowListener()`, this code declares and instantiates an anonymous class that has one method `windowClosing()`.

Java Applets in a Web page


Java Applet
HTML page with the tag to embed a Java applet
Web Browser with Java plugin
User's Computer
Web Server

Java Applets is an outdated technology.
We'll review applets just for historical reasons.

HTML in 10 minutes (start)

```
<!DOCTYPE html>
<HTML>

     <HEAD>
       <TITLE>My First Web Page</TITLE>
     </HEAD>

    <BODY>
         My Tic-Tac-Toe applet is coming soon…
    </BODY>

</HTML>
```

Hyper Text Markup Language (HTML) is not a programming language.

It includes a set of *tags* that you can use with any plain text document to let Web Browsers know how to display the document.

HTML in 10 minutes (cont.)

```
<HTML>

  <HEAD>
    <TITLE>My First Web Page</TITLE>
  </HEAD>

  <BODY>
     My Tic-Tac-Toe applet is coming soon…
    <br>

  In the meantime, listen to <a href="http://americhka.us">Budam's podcasts</a>

  </BODY>

</HTML>
```

The browser will go there

This is a hyperlink

The user will see this

HTML in 10 minutes (end)

```
<HTML>

 <HEAD>
  <TITLE>My First Web Page</TITLE>
 </HEAD>

<BODY>
  My Tic-Tac-Toe applet is coming soon…

 <APPLET code="TicTacToe.class" width=300 height=250 />

 <br>
  In the meantime, listen to <a href="http://americhka.us">Budam's podcasts</a>

  </BODY>

</HTML>
```

Include the Java applet TicTacToe here

Walkthrough 1

Enter the text below in a plain text editor like Notepad or TextEdit.

Save it in the file HelloWorld.html and open this file in your Web browser using the menu File | Open.

```
<HTML>

 <HEAD>
  <TITLE>My First Web Page</TITLE>
 </HEAD>

<BODY>

  My Tic-Tac-Toe applet is coming soon. In the meantime, listen to <a href="http://americhka.us">Budam podcasts</a>
  <p>

  <APPLET code=”TicTacToe.class” width=300 height=250 />
 </BODY>
</HTML>
```

Why don’t you see the TicTacToe applet?

Swing Applet Hello World

```
import javax.swing.JApplet;
import java.awt.Graphics;

public class HelloWorld extends JApplet {

   public void paint(Graphics g){
        g.drawString("Hello World", 50, 100);
   }
}
```

Applet's Callbacks

- **init()** is called when the applet is loaded by the browser's Java Plug-in. It's called only once like constructors in regular Java classes.

- **start()** is called right after the init(). It is called each time when a user returns to the Web page with the applet after visiting another page.

- **paint()** is called when the applet's window needs to be displayed or refreshed after some activity on the screen. For example, the applet is overlapped with some other window and the browser needs to repaint it. This method gets an instance of the Graphics object as an argument.

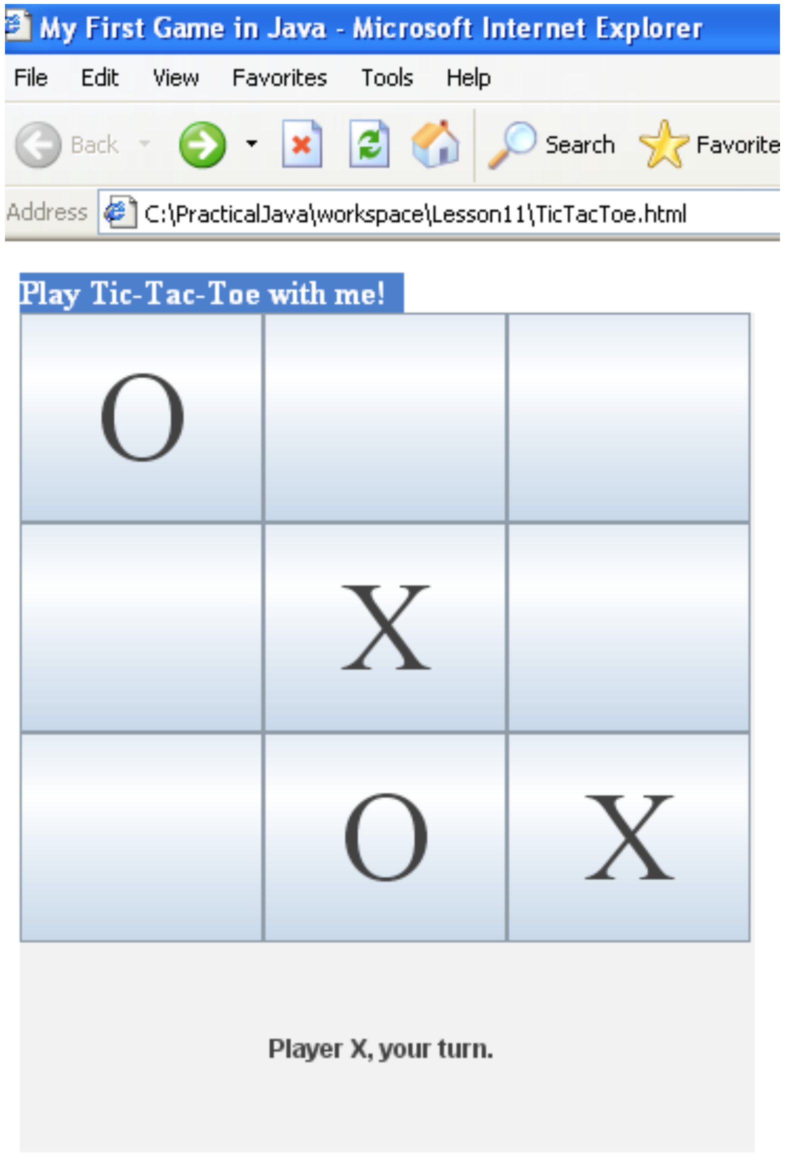
- **stop()** is called when the user leaves the Web page containing the applet.

- **destroy()** – is called when the browser destroys the applet. Write code in this method only if the applet uses some other resources, for example, it holds a connection to the remote computer.

Walkthrough 2

- Create an Eclipse project called MyApplets
- Create and compile an applet class HelloWorld there
- Copy HelloWorld.class into the same directory where HelloWorld.html is located
- Modify the <applet> tag in html file to include HelloWorld.class
- Open HelloWorld.html in the Web browser – your applet should be displayed in the Web page.

Tic-Tac-Toe Applet


My First Game in Java - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorite
Address
C:\PracticalJava\workspace\Lesson11\TicTacToe.html
Play Tic-Tac-Toe with me!
O
X
O
X

Player X, your turn.

Walkthrough 3

- Download the source code from Lesson11 and import it to Eclipse
- Run the program TickTacToeApplet1 in *appletviewer* (right-click, Run As Java Applet)
- Read the code and explain it to yourself
- Review the code of TickTacToe.html
- Run in the TickTacToeApplet in *appletviewer* and review the code with instructor

Web Servers, Servlet Containers, App Servers

- Web Servers just serve static content: HTML Web pages, files, images, media files, binaries, CSS files et al.
  Popular Web servers: Apache, IIS, nginx

- Java Servlet Containers include a JVM, where certain Java programs can run and serve dynamic content to the end users via the Web servers.
  Popular servlet containers: Apache Tomcat, Jetty, Resin

- Java Application Servers include a JVM with Servlet Container, EJB container, and can run all types of Java programs written according to Java EE Specification.
  Popular Java App servers: Web Sphere, WildFly, WebLogic, GlassFish

Configuring Apache Tomcat 7 Server in Eclipse

1. Download and unzip Apache Tomcat from http://tomcat.apache.org/download-70.cgi. Select the Binary Distributions (Core).

2. In Eclipse, go to Java EE perspective, select the menu File | New | Other | Server |Server | Apache | Tomcat v7.0 Server, select Tomcat installation directory and press Finish.
If you don't see Tomcat 7 in the list of Apache servers,
click on "Download additional server adapters".

3. Go to menu Windows | Show View and open the Servers view.
Start Tomcat using the right-click menu.

4. Double-click on the Tomcat entry, select the radio button *Use Tomcat Installation.*
This tells Eclipse to deploy your classes in the original Tomcat directories and not it internal Eclipse directories.

5. Change the content of the Deploy path field from wtpwebapps to webapps.
Press Ctrl-S to save changes. We just want to use the original Tomcat dir webapps.

Manual Deployment to Apache Tomcat 7

1. Open your Window Explorer or Finder and examine
   the tomcat/webapps/ROOT directory. Copy there HelloWorld.html from Walkthrough 1.

5. Open the URL *http://localhost:8080/HelloWorld.html* in your Web browser – the content of HelloWorld.html is displayed

6. Copy *TicTacToe.html and the bin directory* from your Eclipse workspace to */webapps/ROOT*
   Open the URL http://localhost:8080/TicTacToe.html and play your TicTacToe game.

In the future, we will not be doing manual deployments – Eclipse will do it for us.

In the real world projects, someone in your team will write a build script using
one of the build tools (Ant, Maven, Gradle, etc.) that will compile, and copy the required
files into the proper server directory.

Homework

1. Learn HTML basics: http://www.htmldog.com/guides/html/beginner .

3. Install and configure Apache Tomcat 7 as per instructions from the previous slide. For more details visit http://www.coreservlets.com/Apache-Tomcat-Tutorial/

4. Deploy your TicTacToe applet under Apache Tomcat.

5. Study materials from lessons 10 and 11 and and do the Try It assignment from lesson 11.

Additional Read

Unified Modeling Language (UML) Refcard:
http://cdn.dzone.com/sites/all/files/refcardz/rc112-010d-uml.pdf

Google Java Coding Style Standards:
http://google-styleguide.googlecode.com/svn/trunk/javaguide.html