

# Eclipse Plug-in Architecture

EE 564

Lecture 8

Daqing Hou, Winter 2007

# Eclipse plug-in architecture

is built on following techniques

- Java reflection
- Java class loading
- Jar API
- XML processing

# Explain this code

---

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
Connection con = DriverManager.getConnection(  
    "jdbc:oracle:thin:@dbaprod1:1544:SHR1_PRD",  
    username, passwd);
```

# and this!

---

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");    }}
}
```

# Background

---

- Java byte code & class files
  - cf [Java Virtual Machine Specification, 2nd ed.](#)

```
dhou$ hexdump -C Factor.class
00000000 ca fe ba be 00 00 00 31 00 50 07 00 02 01 00 14 |.....1.P.....|
00000010 63 6c 61 72 6b 73 6f 6e 2f 6d 61 74 68 2f 46 61 |clarkson/math/Fal
00000020 63 74 6f 72 07 00 04 01 00 10 6a 61 76 61 2f 6c |ctor.....java/|
00000030 61 6e 67 2f 4f 62 6a 65 63 74 01 00 06 3c 69 6e |ang/Object...<in|
00000040 69 74 3e 01 00 03 28 29 56 01 00 04 43 6f 64 65 |lit>...(V...Code|
00000050 0a 00 03 00 09 0c 00 05 00 06 01 00 0f 4c 69 6e |.....Lin|
00000060 65 4e 75 6d 62 65 72 54 61 62 6c 65 01 00 12 4c |eNumberTable...L|
00000070 6f 63 61 6c 56 61 72 69 61 62 6c 65 54 61 62 6c |ocalVariableTabl|
00000080 65 01 00 04 74 68 69 73 01 00 16 4c 63 6c 61 72 |e...this...Lclar|
00000090 6b 73 6f 6e 2f 6d 61 74 68 2f 46 61 63 74 6f 72 |kson/math/Factor|
000000a0 3b 01 00 12 50 72 69 6d 65 46 61 63 74 6f 72 69 |;...PrimeFactori|
000000b0 7a 61 74 69 6f 6e 01 00 16 28 49 29 4c 6a 61 76 |zation...(I)Ljav|
000000c0 61 2f 75 74 69 6c 2f 48 61 73 68 4d 61 70 3b 01 |a/util/HashMap;.|
000000d0 00 09 53 69 67 6e 61 74 75 72 65 01 00 3e 28 49 |..Signature..>(I|
```

- Java class files contain also data about types (a.k.a meta-data)

# Reflection API

---

- Type info. made available at runtime
- `java.lang.Class`
  - `forName("qualified class name")`
  - `newInstance()`
  - `getConstructor(Class[] paramType)`
  - `getMethod(String name, Class[] paramTypes)`
  - `getField(String name)`

# Reflection API

---

- `java.lang.reflect.Method`
  - `Object invoke(Object, Object[])` throws `IllegalAccessException` - if this `Method` object enforces Java language access control and the underlying method is inaccessible.

# Java class loader

---

- Creates class from Java byte code
- Subclass `java.lang.ClassLoader` to create new loaders
- New loader knows where to load a class
- For each application,
  - JVM provides bootstrap loader
  - optionally, application may define loaders
  - All class loaders form a tree rooted at the bootstrap loader



# java.lang.ClassLoader

---

- java.lang.ClassLoader
  - public loadClass()  
*call parent loader first if failed then findClass()*
  - protected findClass()
  - protected final defineClass()
  - protected findLoadedClass()

# Why class loader?

---

- Division and integration of work
- network computing
- security

# Tomcat's class loaders

- Bootstrap, system, common, catalina, shared, web app loaders
- Each loader knows a location to load classes
- **See** <http://tomcat.apache.org/tomcat-4.1-doc/class-loader-howto.html>

# Whistles and bells

---

- Use XML to do configuration as eclipse manifest files do
- Package classes and resources of plug-ins in jar format
- Load classes using API from `java.util.jar`

# Resources on internet

---

- **Class loading**

<http://java.sun.com/docs/books/tutorial/ext/basics/load.htm>  
↓

- **Jar**

<http://java.sun.com/docs/books/tutorial/deployment/jar/index.html>

- **XML processing**

<http://www.saxproject.org/>