Anno Accademico 2007-2008

Laboratorio di Tecnologie Web
Introduzione ad Eclipse e Tomcat

http://www-lia.deis.unibo.it/Courses/TecnologieWeb0708/

Eclipse – Getting started

> Java SDK

- http://java.sun.com/javase/downloads/index.jsp

> Eclipse IDE

- http://www.eclipse.org/downloads/

> Troubleshooting

- http://www.google.com/

Eclipse – New project...

> Launch Eclipse

- Choose your workspace location

> Welcome... later on

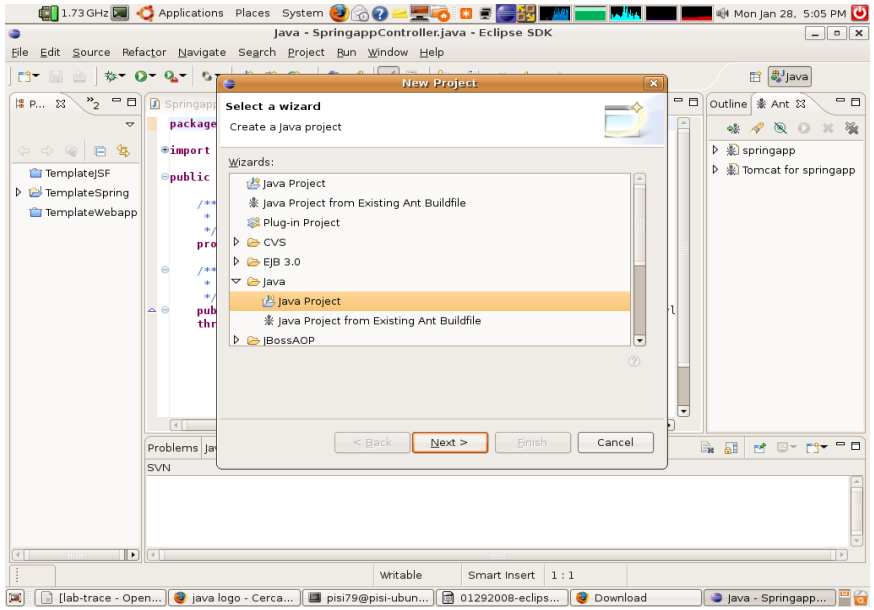
- Help → Welcome

> New project wizard

- Java project → Default options

> Ok: let's code!

- Java project → Default options



- New package.. what name?, new class, static fields and member fields, naming conventions, constructor, scope modifiers

- Comments!

Eclipse – Where the IDE comes in

> What you cannot have on notepad

- autocompletion (e.g., brackets, variable names, generated source)
- language keys highlighting
- different styles for different items
- error notifications → The 'Error' view: ***Look at here, first*!**
- refactor: *"I've changed my mind.. MyClass will become a SuperClass of something else...; and I can do that on packages, too!"*

> Mind the browser

- Eclipse on-line tutorials (e.g., http://eclipsetutorial.sourceforge.net)

Eclipse – IDE features

> Other IDE features

- views
- perspectives
- drop-down menus and contextual ones
- workspace, project and working set
- references and declarations
- ...

> Javadoc

- File → Export → ... Javadoc
- Open *index.html*; open with...

Eclipse – Unit testing

> JUnit3 (Eclipse 3.1) / JUnit4 (Eclipse 3.3) integration

- Test-driven development and Xtreme Programming
- Wizard tool
- JUnit libraries & the build path
  - A chance to see Quick Fix (CTRL+1) in action
- Sample testing

Eclipse – Packaging

> Enabling packaging...

- Project properties
- GUI tool

> "*Actually I can never remember this!*"

- Eclipse Help → Search...
- Let's go googling
- Spend time to learn Ant

Eclipse – Ant within the IDE

> Why Ant

- Human-readable, programmatic way of doing things
- Do tasks as many times you like

> Download and installation

- Already in Eclipse: Ant support, Ant view
- (manual: http://ant.apache.org/manual/index.html)
- Custom libraries for custom-defined tasks (e.g., Tomcat's)

> Prepare the project structure, compile, package it.

- ANT classpath *vs.* Eclipse project classpath
- Environment properties

Towards Java EE applications

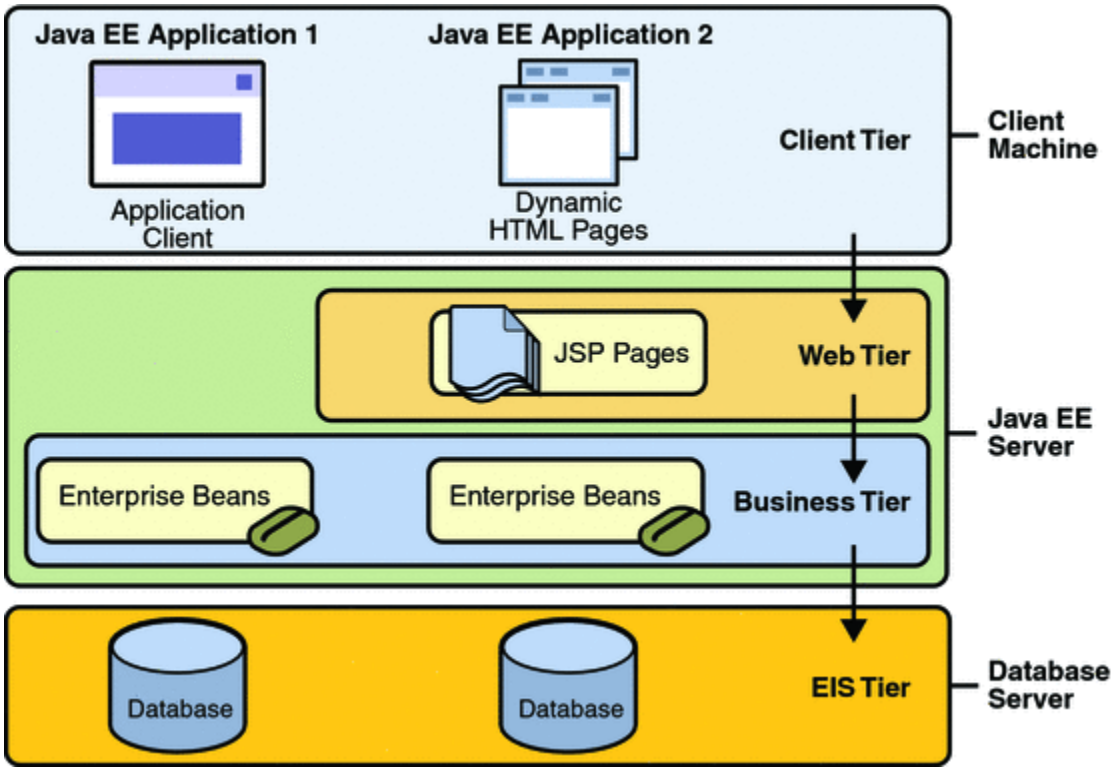
> Straight from the JEE tutorial (http://java.sun.com/javaee/5/docs/tutorial/doc/):

- "*Java EE applications are **made up of components**.*"
- "*A Java EE component is a **self-contained functional software unit** that is **assembled into a Java EE application** and that communicates with other components.*"
- "*Java EE components are written in the Java programming language and are compiled in the same way as any program in the language. The difference between Java EE components and 'standard' Java classes is that Java EE components are assembled into a Java EE application, are verified to be well formed and in compliance with the Java EE specification, and are **deployed to production**, where **they are run and managed by the Java EE server.***"

Java EE components

> The Java EE specification defines the following Java EE components:

- **Application clients** and **applets** are components that run on the client.
- **Java Servlet**, **JavaServer Faces** (JSF), and **JavaServer Pages** (JSP) technology components are web components that run on the server.
- **Enterprise JavaBeans** (EJB) components (enterprise beans) are business components that run on the server.



Java EE containers

> Java EE architecture makes Java EE applications easy to write because **business logic is organized into reusable components**.

> In addition, the Java EE server provides **underlying services** in the form of **a container for every component type**. Because you do not have to develop these services yourself, you are free to concentrate on your biz logic.

> Containers are the interface between a component and the low-level platform-specific functionality that supports the component. Before a web, enterprise bean, or application client component can be executed, it must be assembled into a Java EE module and deployed into its container.

> Containers include services such as:

- security,
- transaction management,
- Java Naming and Directory Interface (JNDI) lookups

Java EE container types

> A JEE server realizes the runtime portion of the Java EE framework.

> The JEE server provides two major types of container:

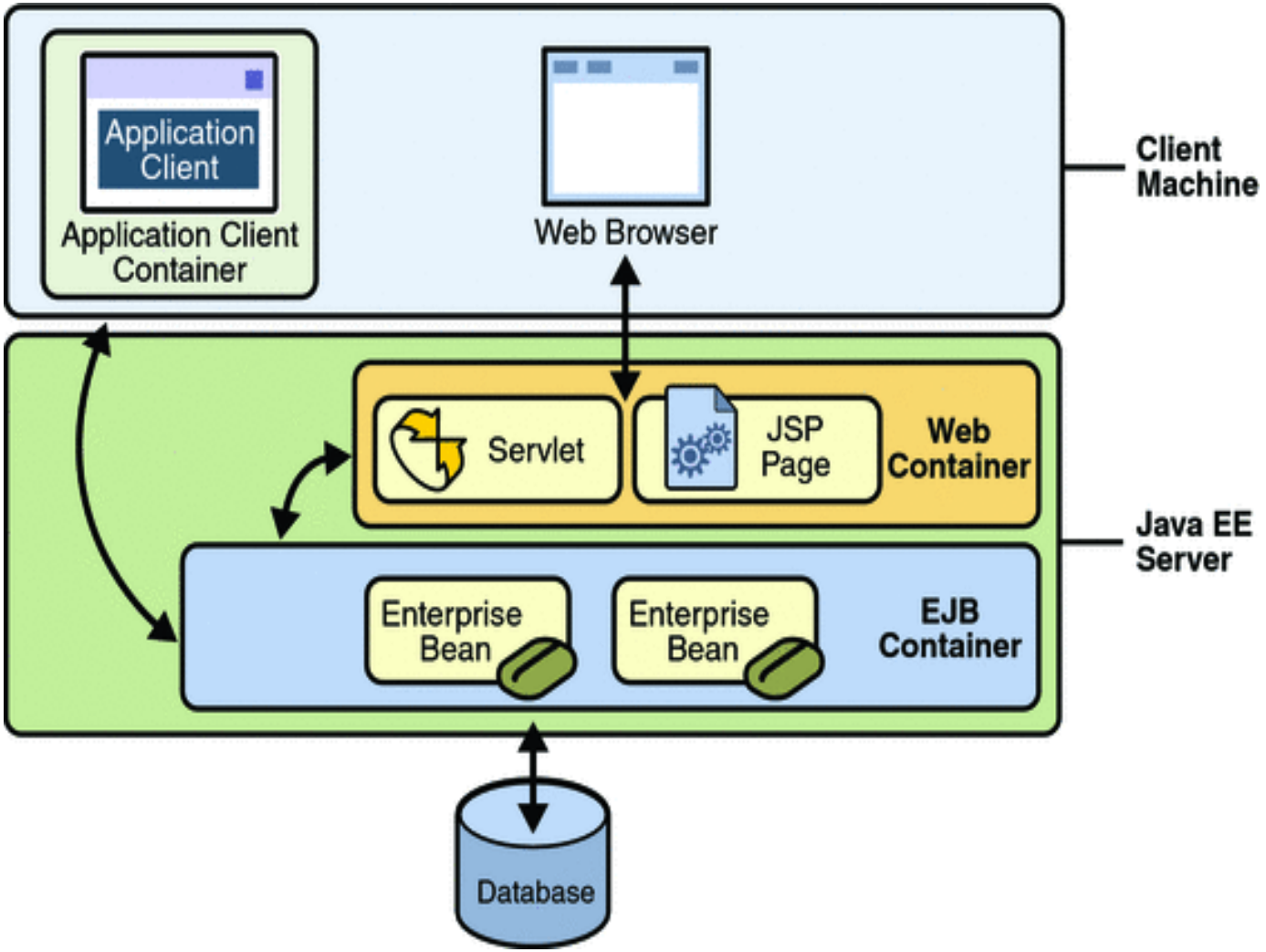
- **Enterprise JavaBeans (EJB) container**: Manages the execution of enterprise beans for Java EE applications. Enterprise beans and their container run on the Java EE server.
- **Web container:** Manages the execution of JSP page and servlet components for Java EE applications. Web components and their container run on the Java EE server.

> Besides:

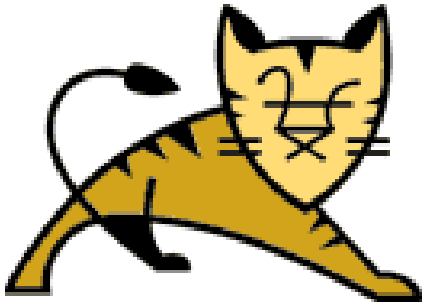
- **Application client container:** Manages the execution of application client components. Application clients and their container run on the client.
- **Applet container:** Manages the execution of applets. Consists of a web browser and Java Plug-in running on the client together.

Java EE container types

> At a glance:



Tomcat – A JEE *Web* server



> Provides **Servlet** and **JSP** containers.

> Just download and explode it

- http://tomcat.apache.org

> Tomcat structure

- **webapps**: where to deploy your web applications
- **conf:** configuration files (e.g., configure tomcat users: the admin)
- **log:** log files will be placed here

> Launch Tomcat

- JRE location
- http://localhost:8080. Ok. It works.

> Commanding Tomcat from Ant

The very first webapp

> Eclipse project structure

- **lib:** additional webapp libraries (we will use this for database connectors, JSF implementation libraries, etc..)
- **war:** resembling the webapp archive structure
  - XML descriptors
  - resources
  - classes
- ...

> Ant tomcat libraries

- Define tomcat tasks by the catalina Ant librarias