Java项目讲座

时间：1011-12-09 15:00 ~ 17:00（2小时）

内容：

1．(10)java运行环境介绍

2．(15)最简单web项目结构，纯jsp组成，tomcat中运行

3．(10)组件的编写，jar包生成

4．(10)重写web项目，依赖jar，使用servlet，生成war

5．(15)IDE的使用SpringSource Tool Suite（Eclipse）

6．休息

7．(20)当前流行的项目管理工具Maven介绍

8．(20)Maven方式创建项目，一个完整项目介绍SpringMVC+Spring+ibatis

9．(10)Maven编译，测试，打包，发布

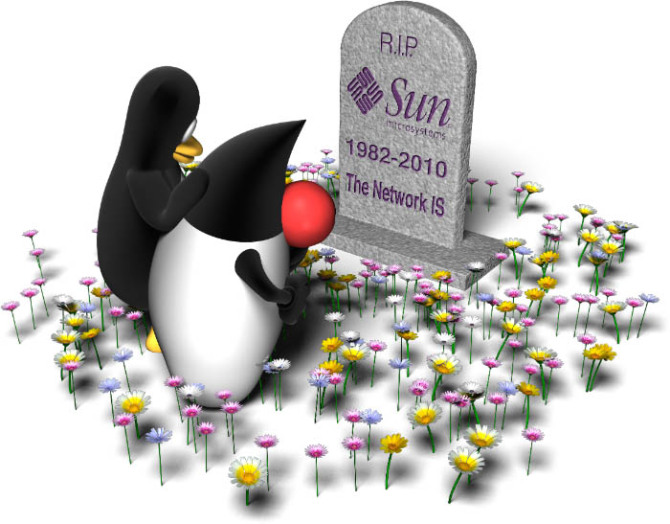
补充：

时间充裕的话，添加以下内容，会出现在ppt中

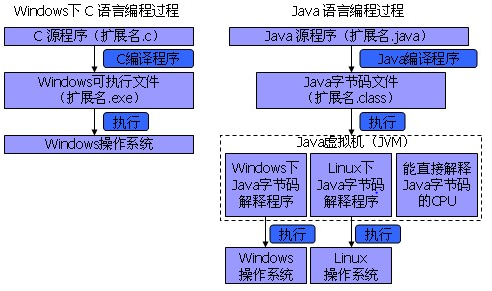
添加常用服务器了解： jboss，tomcat，jetty， websphere，weblogic，

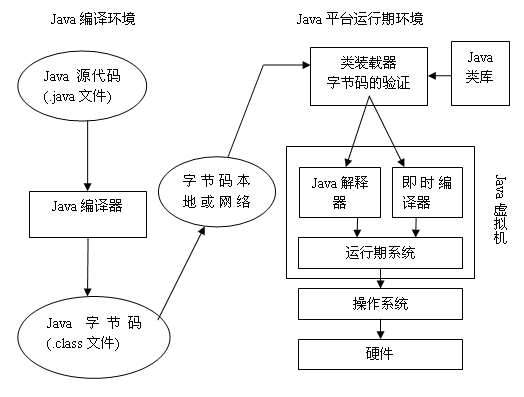
Java学习重点与参考内容

# java介绍



## java工作原理





## Java版本

J2SE:

Java 2 Standard Edition(标准版)

Java语言核心的类,数据库连接、接口定义、输入/输出、网络编程

J2EE:

Java 2 Enterprise Edition(企业版)

包含servlet、JSP、XML、EJB、事务控制



J2ME:

Java 2 Micro Edition(微缩版)

针对嵌入式系统，包括呼机、智能卡、手机、PDA、机顶盒开发

Java Card:开发可以植入IC智能卡中的应用程序。

## Java术语

### JRE

Java RunTime Envirment(JAVA运行环境)

### JDK

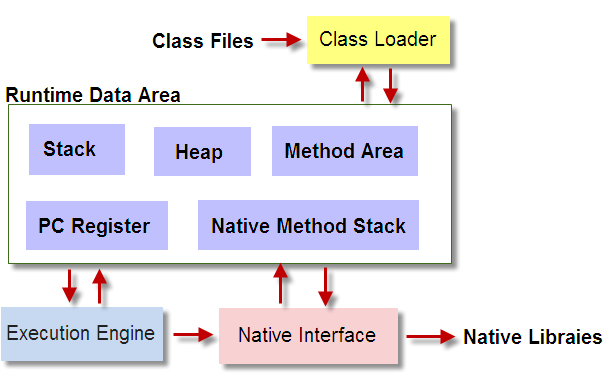
Java Developer’s Kit (Java开发者工具包),

也叫J2SDK(Java 2 Software Development Kit)，包含：文档，编译工具、运行工具以及执行程序的环境(即JRE)。

### JVM

Java Virtual Machine(Java虚拟机，虚拟的计算机)

JVM专题<http://developer.51cto.com/art/201001/176550.htm>



### GC

Garbage Collection(垃圾回收器)

## JDK安装

### 下载安装：

[http://www.oracle.com](http://www.oracle.com/) ：jdk(J2SE)，weblogic，netBeans，mysql，GlassFish

<http://www.oracle.com/technetwork/java/javase/downloads/index.html> jdk-6u29-windows-i586.exe

下载下来直接安装，也可以从别人那里拷贝过来！

C:\java\jdk1.6.0\_29

路径中尽量不要有空格，中文等特殊字符

不需要另外安装jre

### 环境变量

Winxp 我的电脑 / 属性 / 高级 / 环境变量

Win 7 我的电脑 / 属性 / 高级系统设置 / 环境变量

用户变量：

JAVA\_HOME = C:\java\jdk1.6.0\_29

path = %JAVA\_HOME%\bin;%M2\_HOME%\bin;

验证

java -version

### 目录介绍

bin Java 平台使用的工具和库的可执行文件及 DLL

demo 程序示例，演示

include C 语言头文件

lib 开发工具使用的文件。

Jre JDK 开发工具所使用的 Java 运行时环境的根目录

src.jar API源文件

### JDK基本命令

常用命令

javac 编译

java 运行

jar 打包命令

javadoc 生成文档

其他命令

appletviewer 、 jdb 、 javah 、 javap 、 extcheck

### javac

javac 是 JAVA 语言的编译器。基本语法是：

javac [ 选项 ] 源文件名

常用的选项是：

**-cp**

**-classpath classpath** ： 定义 javac 搜索类的路径。

它将覆盖默认的 classpath 环境变量的设置：一般格式是： .;<your\_path>

例如： .;c:/java/doc/classes;c:/tools/java/classes

**-d directory** : 指明编译生成的类所存放的根目录。

例如： javac -d <my\_dir> MyProgram.java

这样将 MyProgram.java 编译后所产生的 MyProgram.class 存放在 my\_dir 目录中。

### jar

jar命令同linux的tar

jar cf demo.jar com

### 打包jar的manifest.mf

jar cvfm sms.jar MANIFEST.MF .

.表示当前目录classes下的

MANIFEST.MF

Manifest-Version: 1.0

Ant-Version: Apache Ant 1.6.5

Created-By: SMS

Main-Class: com.seaway.smgp.SMSMain

Revision: 5100

Class-Path: lib/bcprov-jdk14-133.jar lib/c3p0-0.9.1.2.jar lib/castor-1.3-core.jar lib/castor-1.3-xml.jar lib/commons-collections.jar lib/commons-lang-2.4.jar lib/commons-logging-1.0.4.jar lib/dom4j-1.6.1.jar lib/hibernate-3.2.6.ga.jar lib/jta-1.0.1B.jar lib/log4j-1.2.8.jar lib/mina-core-2.0.0-M4.jar lib/mina-filter-compression-2.0.0-M4.jar lib/ojdbc14.jar lib/slf4j-api-1.5.6.jar lib/slf4j-jdk14-1.5.6.jar lib/smproxynew.jar lib/smproxy12.jar lib/sw\_v\_2\_0.jar

参见：<http://blog.csdn.net/javadxz/article/details/6524282>

|  |
| --- |
| 1.在命令行中输入“jar -help”，就k可看到jar的详细用法了。  2.示例：  1）将两个class文档归档到一个jar文件中：jar -cvf XX.jar A.class B.class  2 )使用清单文件Manifest.mf将dir目录下的所有文件归档到一个jar中：jar -cvfm XX.jar Manifest.mf dir/\*  manifest 文件的格式很简单，每一行都是 " 名－值 " 对应的：  属性名开头，接着是 ":" ，然后是属性值，每行最多72个字符，如果需要增加，你可以在下一行续行，续行以空格开头，以空格开头的行 都会被视为前一行的续行。  使用 m 选项，把指定文件名的manifest文件传入，例如  jar cvfm myapplication.jar myapplication.mf [-C]classdir  Manifest.mf的详细用法（转）  现在我们来体验一下manifest文件的作用，如果现在我们有一个Java 应用程序打包在myapplication.jar中， main class为 com.example.myapp.MyAppMain ，那么我们可以用以下的命令来运行  java -classpath myapplication.jar com.example.myapp.MyAppMain  这显然太麻烦了，现在我们来创建自己的manifest文件，如下：  Manifest-Version: 1.0  Created-By: JDJ example  Main-Class: com.example.myapp.MyAppMain  这样我们就可以使用如下的命令来运行程序了：（明显简单多了，也不会造成无谓的拼写错误）  java -jar myapplication.jar  管理JAR的依赖资源  很少Java应用会仅仅只有一个jar文 件，一般还需要 其他类库。比如我的应用程序用到了Sun 的 Javamail classes ，在classpath中我需要包含activation.jar 和 mail.jar,这样在运行程序时,相比上面的例子,我们要增加一些:  java -classpath mail.jar:activation.jar -jar myapplication.jar  在不同的操作系统中,jar包间的分隔符也不一样，在UNIX用“:”，在 window中使用 “;”，这样也不方便  同样，我们改写我们的manifest文件，如下  Manifest-Version: 1.0  Created-By: JDJ example  Main-Class: com.example.myapp.MyAppMain  Class-Path: mail.jar activation.jar  （加入了Class-Path: mail.jar activation.jar，用空格分隔两个jar包）  这样我们仍然可以使用和上例中相同的命令来执行该程序：  java -jar myapplication.jar  Class-Path属性中包含了用空格分隔的jar文件，在这些jar文件名中要对特定的字符使用逃逸符，比如空格，要表示成"%20"，在路径的表示中，都采用“/” 来分隔目录()，无论是在什么操作系统中，(即使在window中)，而且这里用的是相对路径（相对于本身的JAR文 件）：  Manifest-Version: 1.0  Created-By: JDJ example  Main-Class: com.example.myapp.MyAppMain  Class-Path: ext/mail.jar ext/activation.jar  Multiple Main Classes（多主类）  还有一种Multiple Main Classes情况，如果你的应用程序可能有命令行版本 和GUI版本，或者一些不同的应用却共享很多相同的代码，这时你可能有多个Main Class，我们建议你采取这样的策略：把共享的类打成lib包，然后把不同的应用打成不同的包，分别标志主类：如下  Manifest for myapplicationlib.jar:  Manifest-Version: 1.0  Created-By: JDJ example  Class-Path: mail.jar activation.jar  Manifest for myappconsole.jar:  Manifest-Version: 1.0  Created-By: JDJ example  Class-Path: myapplicationlib.jar  Main-Class: com.example.myapp.MyAppMain  Manifest for myappadmin.jar:  Manifest-Version: 1.0  Created-By: JDJ example  Class-Path: myapplicationlib.jar  Main-Class: com.example.myapp.MyAdminTool  在myappconsole.jar 和 myappadmin.jar的manifest文件中分别注明各自的 Main Class  Package Versioning  完成发布后，如果使用者想了解 ，哪些代码是谁的？目前是什么版本？使用什么版本的类库？解决的方法很多 ，manifest提供了一个较好的方法，你可以在manifest文件中描述每一个包的信息。  Java 秉承了实现说明与描述分离的原则，package 的描述 定义了package 是什么，实现说明 定义了谁提供了描述的实现，描述和实现包含 名、版本号和提供者。要得到这些信息，可以查看JVM的系统属性（使用 java.lang.System.getProperty() ）  在manifest文件中，我可以为每个package定义描述和实现版本，声明名字，并加入描述属性和实现属性，这些属性是  Specification-Title  Specification-Version  Specification-Vendor  Implementation-Title  Implementation-Version  Implementation-Vendor  当要提供一个类库或编程接口时，描述信息显得是很重要，见以下范例：  Manifest-Version: 1.0  Created-By: JDJ example  Class-Path: mail.jar activation.jar  Name: com/example/myapp/  Specification-Title: MyApp  Specification-Version: 2.4  Specification-Vendor: example.com  Implementation-Title: com.example.myapp  Implementation-Version: 2002-03-05-A  Implementation-Vendor: example.com  Package Version 查询  在manifest文件中加入package描述后，就可以使用Java提供的java.lang.Package class进行Package 的信息查询，这里列举3个最基本的获取package object的方法  1.Package.getPackages():返回系统中所有定义的package列表  2.Package.getPackage(String packagename):按名返回package  3.Class.getPackage():返回给定class所在的package  使用者这方法就可以动态的获取package信息.  需要注意的是如果给定的package中没有class被加载,则也无法获得package 对象  Manifest 技巧  总是以Manifest-Version属性开头  每行最长72个字符，如果超过的化，采用续行  确认每行都以回车结束，否则改行将会被忽略  如果Class-Path 中的存在路径，使用"/"分隔目录，与平台无关  使用空行分隔主属性和package属性  使用"/"而不是"."来分隔package 和class ,比如 com/example/myapp/  class 要以.class结尾，package 要以 / 结尾 |

# java命令行程序

## demo01-HelloWord

demo01 程序基本结构

|  |
| --- |
| public class Demo01{  public static void main(String[] args){  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("类名：" + Demo01.class);  System.out.println("时间：" + new java.util.Date());  int sum = 0;  for(int i=1; i<=100;i++){  sum = sum + i;  }  System.out.println("sum(1..100)="+sum);  }  } |
| javac \*.java //编译  java Demo01 //运行 |

一个文件一个类

一个文件可以定义多个类，但只能有一个public类，public类必须和文件名相同

文件名类名关系

包

完整类名：com.jrj.train.cmd.Demo04

## demo02-ListProcess

ListProcess 调用系统命令

|  |
| --- |
| **package** com.jrj.train.demo01;  **import** java.io.\*;  **public** **class** ListProcess {  **public** **static** **void** main(String[] args) **throws** IOException {  Runtime runtime = Runtime.*getRuntime*();  //String cmds[] = { "cmd", "/c", "tasklist" };  String cmds[] = {"tasklist" };  Process proc = runtime.exec(cmds);  InputStream inputstream = proc.getInputStream();  InputStreamReader inputstreamreader = **new** InputStreamReader(inputstream);  BufferedReader bufferedreader = **new** BufferedReader(inputstreamreader);  String line;  **while** ((line = bufferedreader.readLine()) != **null**) {  System.*out*.println(line);  }  }  } |

## java-直接运行的jar

示例程序demo03-FivaGame

1 普通jar

|  |
| --- |
| javac \*.java -d classes //编译所有程序到classes目录  jar -cf fivagame.jar . //打包  java -cp fivagame.jar PlayChess //运行 |

2 自运行jar

准备清单信息

manifest.txt 最后一行要有回车

|  |
| --- |
| Main-Class: PlayChess |

也可以直接写入fivegame/META-INF/MANIFEST.MF

打包命令

|  |
| --- |
| jar -cfm fivagame.jar manifest.txt \* //打包  或 jar -cmf manifest.txt myjar.jar .  java -jar fivagame.jar //运行 |

## jar组件

示例：demo04-utils

|  |
| --- |
| mkdir classes  javac -encoding utf-8 \*Utils.java -d classes //编译.  javac Main.java //错误  javac -cp classes Main.java  jar cf jrj-utils.jar -C classes . //打包  javac -cp classes Main.java //编译Main  java –cp .;classes Main //运行Main  或  javac -cp jrj-utils.jar Main.java //编译Main  java -cp .;jrj-utils.jar Main //运行Main |

# web01

tomcat安装参见常见服务器

## 项目结构

webapps下创建如下项目结构

|  |
| --- |
| apache-tomcat-7.0.23  |\_webapps  |\_web01  |-WEB-INF  | |-classes  | |-lib  | |-web.xml  |-index.jsp |

## jsp

Index.jsp

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <%@ page import=*"java.util.Date"* %>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>login</title>  </head>  <body>  <!-- html注释 -->  <%-- script注释 --%>  <%-- 声明变量 --%>  <%! Date date = **null**;%>      <%  //变量赋值  date = **new** Date();  String name = "hch";    out.print("Hello! " + name);  %>    <br>    <%--表达式输出 --%>  <%=date%>  </body>  </html> |

## servlet

RequestInfo.java (Source:Tomcat / Servlet Examples / Request Info)

|  |
| --- |
| package com.jrj.demo.web01.servlet;  import java.io.\*;  import javax.servlet.\*;  import javax.servlet.http.\*;  public class RequestInfo 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("<body>");  out.println("<head>");  out.println("<title>Request Information Example</title>");  out.println("</head>");  out.println("<body>");  out.println("<h3>Request Information Example</h3>");  out.println("Method: " + request.getMethod());  out.println("<br> ");  out.println("Request URI: " + request.getRequestURI());  out.println("<br> ");  out.println("Request URL: " + request.getRequestURL());  out.println("<br> ");  out.println("ParameterMap: " + request.getParameterMap());  out.println("<br> ");  out.println("Protocol: " + request.getProtocol());  out.println("<br> ");  out.println("PathInfo: " + request.getPathInfo());  out.println("<br> ");  out.println("Remote Address: " + request.getRemoteAddr());  out.println("<br> ");  out.println("</body>");  out.println("</html>");  }  /\*\*  \* We are going to perform the same operations for POST requests  \* as for GET methods, so this method just sends the request to  \* the doGet method.  \*/  public void doPost(HttpServletRequest request, HttpServletResponse response)  throws IOException, ServletException  {  doGet(request, response);  }  } |
| javac -cp servlet-api.jar \*.java -d classes |

Servlet编译

将tomcat/lib下的servlet-api.jar加入到classpath中

javac -classpath servlet-api.jar RequestInfo.java -d .

生成的类放置在WEB-INF/classes下

或

javac -classpath servlet-api.jar RequestInfo.java -d WEB-INF/classes

## web.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <web-app xmlns="http://java.sun.com/xml/ns/javaee"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0">  <servlet>  <servlet-name>RequestInfo</servlet-name>  <servlet-class>com.jrj.demo.web01.servlet.RequestInfo</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>RequestInfo</servlet-name>  <url-pattern>/RequestInfo</url-pattern>  </servlet-mapping>  <welcome-file-list>  <welcome-file>index.jsp</welcome-file>  </welcome-file-list>  </web-app> |

启动tomcat访问

http://localhost:8080/web01

完整web01参见代码web01

## servlet-jar

classes

application lib

Tomcat lib

## war

jar -cf web01.war .

## 其他

JSP (Java Server Pages)

语法

|  |  |
| --- | --- |
| 指令Directives | <%@ directive %>  <%@ page import="java.util.Date" %>  <%@ page errorPage="errorPage.jsp"%> |
| 声明Declarations | <%! declaration %> |
| 表达式Expressions | <%= expression %> |
| 脚本Scriptlet | <% code fragment %> |
| 注释Comments | <%-- comment --%> 另Html注释； java语言的注释 |

JSP程序设计从入门到精通1-6章(上)

<http://www.docin.com/p-59429591.html>

Servlet:web核心

servlet3新特性

# IDE

Eclipse， NetBeans， IDEA，Jbuilder

## Eclipse

Eclipse | MyEclipse | Spring STS

下载：<http://www.eclipse.org/downloads/>

Eclipse IDE for Java Developers 普通开发工具

Eclipse IDE for Java EE Developers 支持web开发

参考<http://down.51cto.com/zt/124>

## Eclipse-STS

安装了SpringSource Tool Suite的Eclipse，支持maven，

<http://www.springsource.com/products/eclipse-downloads>

<http://www.springsource.com/downloads/sts>

## Other

Myeclipse

NetBeans

IDEA

JDeveloper

JBuilder

NetBeans介绍

<http://developer.51cto.com/art/201105/259431.htm>

Eclipse广告

<http://developer.51cto.com/art/201006/207407.htm>

# Eclipse-使用

## Workspace

工作空间

## 透视图

JavaEE透视图

Java透视图 常用

debug透视图 常用

## 操作

创建项目

导入项目

### 设置workspace字符集

Window / Preferences / General / Workspace

UTF-8

**JSP字符集**

Web / JSP File

### 设置JDK

Window / Preferences / Java / Installed JRSs

添加jdk为新的运行时环境，

### 其他操作

Java / Code Templates 注释末班

Java / Templates java代码模板

Java / compile jdk版本

编码设置

行号显示

快捷键 设置：General | Keys

Alt + space 提示

Ctrl + 鼠标 代码跟踪

显示行号 右键

ALT + / 提示

Ctrl + shift + R 查找资源(resource,文件)

Ctrl + shift + T 查找类

Eclipse快捷键总结

http://bbs.51cto.com/thread-638701-1.html

## Eclipse-tomcat集成

Window | Preferences | Server | Runtime Environments

# 服务器

Java服务器对比<http://special.e800.com.cn/070902/index.html>

## Tomcat

看一看：http://apache.org

### 下载安装：

http://tomcat.apache.org/download-70.cgi

apache-tomcat-7.0.23-windows-x86.zip

解压：D:\0\_server\ apache-tomcat-7.0.23

### 启动：

apache-tomcat-7.0.23\bin\ startup.bat

访问：http://localhost:8080/ 即可访问tomcat的欢迎界面

### 目录结构：

|  |
| --- |
| apache-tomcat-7.0.23  ├─bin 启动和关闭Tomcat的脚本文件  ├─conf 全局配置文件，其中最重要的是server.xml和web.xml  ├─lib 存放Tomcat服务器和所有web应用都能访问的JAR  ├─logs 存放Tomcat的日志文件  ├─temp 存放临时文件  ├─webapps 部署目录  │ ├─docs  │ ├─examples  │ ├─host-manager  │ ├─manager  │ └─ROOT  └─work 工作目录，各种由jsp生成的servlet文件放在这个目录下  └─Catalina  └─localhost  ├─docs  ├─examples  ├─host-manager  ├─manager  └─\_ |

### 管理界面：

### 项目发布

1直接拷贝到webaoos

2 上传war

3 外连接

## Resin

Resin是Caucho公司的产品，非常流行,支持Servlet和JSP的服务器，速度非常快。

Resin本身包含了一个支持HTML的Web服务器，显示静态内容的能力也毫不逊色，因此许多网站都使用Resin服务器构建。

## Jetty

<http://docs.codehaus.org/display/JETTY/Downloading+Jetty>

<http://dist.codehaus.org/jetty/>

## JBoss(Red Hat)

JBoss 遵从JavaEE规范的、开源、纯Java的EJB服务器。其核心服务仅是提供EJB服务器，不包括Servlet和JSP的Web容器，不过它可以和Tomcat完美结合。

下载：http://www.jboss.org/jbossas/downloads/

jboss-as-7.0.2.Final.zip

解压：

启动：bin/standalone.bat

部署：standalone/deployments/web01.war

管理界面：<http://localhost:8080/> / Administration Console

Depliyment / Manage Deployments

参考：

<http://jbosscn.iteye.com/category/162889> jboss系列文档

<http://blog.csdn.net/yingevil/article/details/6906811> jboss7 - 1,2,3,4

<http://bluedash.iteye.com/blog/1282169> maven 创建 部署

## WebLogic

WebLogic是BEA-SUN-Oracle

支持企业级的、多层次的和完全分布式的Web应用，

并且服务器的配置简单、界面友好。提供Java平台所拥有的一切应用服务器的用户来说。

## Webshpere

收费

# 学习重点与参考资料

J2SE：

JDK配置，虚拟机，GC，classloader

标识符，关键词，类型，运算符，表达式，常量

数组，对象，接口，包及访问权限

多线程，IO，集合类collections，工具类util，泛型，反射，StringBuffer

分支，循环，if else switch for while do-while break continue

异常try catch finally throws

国际化i18n

J2EE

DB ：JDBC， mybatis， Hibernate

MVC：

Struts

springmvc

WEB：

jsp + EL + jstl + servlet

cookie + session

javascript + css + ajax

OOAD

UML

CMS

Spring：

IOC DI AOP

BeanFactory ApplicationContext WebApplicationContext

ORM

## 专题：

盘点四大顶级Java编程工具

<http://developer.51cto.com/art/201011/232351.htm>

Java编程之四大名著

<http://developer.51cto.com/art/201104/255281.htm>

## 技术参考网站

<http://www.51cto.com/>

<http://www.infoq.com/>

<http://www.iteye.com/>

<http://www.cnblogs.com/> 博客园

<http://www.csdn.net/>

<http://www.oschina.net>

<http://www.oschina.net/code/explore> 开源项目源码阅读

http://community.csdn.net/ 社区

http://www.itpub.net/

http://tech.sina.com.cn/ 新闻

http://bbs.chinaunix.net/forum/

http://bbs.chinajavaworld.com/index.jspa

http://www.ibm.com/developerworks/cn/java/

# virtualbox虚拟机的搭建：

下载：<https://www.virtualbox.org/wiki/Downloads>

资料：

**MLDN魔乐全新JAVA核心课堂**

http://www.verycd.com/topics/2742618/