Java技术体系图

Java技术路线图

从前文中我们已经了解到，Java SE中包含了Java的基础技术，Java EE中包含了网络应用开发的各种技术，Java ME中包含了移动设备应用开发的各种技术，且Java SE是Java EE和Java ME的基础，所以对于Java技术的学习，有两条学习路线：

Java SE→Java EE：主要进行网络应用开发，是Java技术学习的一条完整路线，是基于Java EE的企业应用开发，是目前Java应用的主要方向。

Java SE→Java ME：主要进行无线应用开发，该技术路线相对简单一些。

Java SE -> Java EE的技术由低到高形成了一个完整的体系，是Java/Java EE工程师成长必备的技术体系，下面我们就根据Java SE -> Java EE的技术体系制定一套完整的学习路线。

**1、Java EE技术分类**

根据Java EE技术的分层，我们可以将Java EE技术分为以下3类。

（1）Web应用技术

JDBC：JDBC（Java DataBase Connectivity）是一个基于标准SQL的数据库访问接口，主要用来连接数据库和直接调用SQL命令执行各种SQL语句和存储过程。JDBC相当与Microsoft平台中的ODBC。

Servlet：Servlet是Java平台上的CGI技术。它在服务器端运行，用以动态生成Web页面。与传统的CGI和许多其他类似CGI的技术相比，Java Servlet具有更高的效率，它依靠线程的方式来支持并发访问。

JSP（EL/JSTL）：JSP（Java Server Page）是一种实现普通静态HTML和动态页面输出混合编码的技术，类似Microsoft ASP、PHP等技术。它借助内容和外观表现的分离，使得Web页面开发可以方便地划分给页面设计人员和程序员。在运行时态，JSP将会被首先转换成Servlet，它的效率和功能与Servlet相比没有差别。

JavaBean：封装业务逻辑的组件，可以实现业务组件的重用。

JSF：JSF（JavaServer Faces）是一种专用于构建Java Web应用的页面组件技术，它集成了Servlet、JSP、JavaBean、EL、JSTL和自定义标签等各种页面技术，方便绘制丰富的页面组件。

（2）企业级应用技术

EJB：EJB定义了一组可重用的组件Enterprise Beans。开发人员可以利用这些组件，像搭积木一样建立分布式应用。EJB为它的开发人员代管了诸如安全性、远程连接、生命周期管理及事务管理等技术环节，简化了商业逻辑的开发。

JNDI：JNDI（Java Naming and Directory Interface）用于定位分布式系统中的各种对象，这些对象包括EJB、数据库驱动、JDBC数据源及消息连接等，还可以使用JNDI访问各种特定的目录服务，如LDAP、NDS和DNS等。

JMS：JMS消息服务是一组Java应用接口，它提供创建、发送、接收、读取消息的服务，使得Java应用能够和各种消息中间件进行通信，这些消息中间件包括IBM MQ-Series、Microsoft MSMQ及纯Java的SonicMQ。

JavaMail：提供了一整套模拟邮件系统的抽象类。通过JavaMail，可以创建邮件或消息应用程序。

JAF：JavaMail利用JavaBeans Activation Framework（JAF）来处理MIME-编码的邮件附件。MIME的字节流可以被转换成Java对象，或者转换自Java对象。

JTA：JTA提供了Java EE中处理事务的标准接口，它支持事务的开始、回滚和提交。同时在Java EE平台上提供一个JTS（Java Transaction Service）作为标准的事务处理服务，开发人员可以使用JTA来使用JTS。

JACC：JACC（Java Authorization Service Provider Contract for Containers）在Java EE应用服务器和特定的授权认证服务器之间定义了一个连接的协约，以便将各种授权认证服务器插入到Java EE产品中去。

JAAS：JAAS（Java Authentication and Authorization Service）实现了一个Java版本的标准Pluggable Authentication Module（PAM）框架。它可用来进行用户身份的鉴定，还能通过对用户进行授权，实现基于用户的访问控制。

JCA：JCA（Java EE Connector Architecture）是Java EE体系架构的一部分，它提供了一套连接各种企业信息系统（包括ERP、SCM、CRM等）的体系架构，开发商只需要开发一套基于JCA的连接适配器，就能够在任何Java EE服务器中连接并使用它。

JMX：JMX（Java Management Extensions）致力于解决分布式系统管理的问题，它是一种应用编程接口、可扩展对象和方法的集合体，可以跨越各种异构操作系统平台、系统体系结构和网络传输协议，开发无缝集成的面向系统、网络和服务的应用。

（3）Web服务技术

JAX-RPC：通过使用JAX-RPC（Java API for XML-based RPC），已有的Java类或Java应用都能够被重新包装，并以Web Services的形式发布。JAX-RPC提供了将RPC参数（in/out）编码和解码的API，使开发人员可以方便地使用SOAP消息来完成RPC调用，EJB应用也可以使用JAX-RPC来包装成Web服务。

SOA（SAAJ）：SAAJ（SOAP with Attachemnts API for Java）是JAX-RPC的一个增强，为进行低层次的SOAP消息操纵提供了支持。

JAXR：JAXR（Java API for XML Registries）提供了与多种类型注册服务进行交互的API。JAXR运行客户端访问与JAXR规范相兼容的Web Servcices，这里的Web Services即为注册服务。一般来说，注册服务总是以Web Services的形式运行的。

**2、Java技术学习路线图**

Java EE技术，从简单到复杂依次包括：Web应用技术、企业级应用技术、Web服务技术，除了这些技术之外，还有一些开源的框架技术，包括Web框架Struts/WebWork/Tapstry，业务层框架Spring，持久层框架Hibernate/iBATIS，这些也都是Java EE的应用技术。

这些技术虽然琳琅满目，但是也是层层递进，逐步形成了Java EE工程师的学习技术路线。除了Java EE技术之外，还有Java基础技术、数据库基础、网页开发技术需要在前期进行掌握。在学习了所有的Java基础、Java EE技术之后，再付诸于实践，你就达到了架构师的水平，进而走上技术管理的道路，可以达到技术道路的峰顶——CTO。

根据Java技术学习路线图，每一步所需掌握的课程有：

**Java程序员**

高级特性

反射、泛型、注释符、自动装箱和拆箱、枚举类、可变参数、可变返回类型、增强循环、静态导入

核心编程

IO、多线程、实体类、集合类、正则表达式、XML和属性文件

图形编程

AWT（Java2D/JavaSound/JMF）、Swing、SWT、JFace

网路编程

Applet、Socket/TCP/UDP、NIO、RMI、CORBA

Java语法基础

类、抽象类、接口、最终类、静态类、匿名类、内部类、异常类、编码规范

Java开发环境

JDK、JVM、Eclipse、Linux

Java核心编程技术

Java，设计而又非常精巧的语言。学习Java，须从Java开发环境开始，到Java语法，再到Java的核心API。

1.Java开发入门：Java开发环境的安装与使用，包括JDK命令、EclipseIDE、Linux下Java程序的开发和部署等。

2.Java语法基础：基于JDK和Eclipse环境，进行Java核心功能开发，掌握Java面向对象的语法构成，包括类、抽象类、接口、最终类、静态类、匿名类、内部类、异常的编写。

3.Java核心API：基于JDK提供的类库，掌握三大核心功能：

A. Java核心编程: 包括Java编程的两大核心功能——Java输入/输出流和多线程，以及常用的辅助类库——实体类、集合类、正则表达式、XML和属性文件。

B. Java图形编程：包括Sun的GUI库AWT（Java2D、JavaSound、JMF）和Swing，IBM和GUI库SWT和Jface;

C. Java网路编程：Applet组件编程，Socket编程，NIO非阻塞Socket编程、RMI和CORBA分布式开发。

4.Java高级特性：掌握JDK1.4、JDK5.0、JDK6.0中的Java高级特性，包括反射、泛型、注释，以及java高级特性——自动装箱和拆箱、枚举类、可变参数、可变返回类型、增强循环、静态导入等。

**Java初级、中级软件工程师**

JSF框架开发技术

配置文件（页面导航、后台Bean）、JSF组件库（JSF EL语言、HTML标签、事件处理、）、JSF核心库（格式转换、输入验证、国际化）

Javaweb核心开发技术

开发环境（Eclipse、Linux）

三大组件（JSP、JavaBean、Servlet）

扩展技术（EL、JSTL、Taglib）

网页开发技术

HTML、XML、CSS、JavaScript、AJAX

数据库设计技术

SQL、MySql、Oracle、SQLServer、JDBC

Web服务器（Tomcat/Jetty/Resin/JBossWeb）

JavaWeb核心技术：

JavaWeb项目开发的全过程可以分解为：

网页开发+数据库设计——>JavaWeb项目开发，其中，javaWeb由6项基本技术组成：JSP+JavaBean+Servlet+EL+JSTL+Taglib，而JSF正是将这6种技术进行有机结合的技术框架。

**Java软件高级工程师**

四种经典架构SSH1、SSI1、SSH2、SSI2

Struts1表现层框架

入门配置、核心组件、标签库、国际化、数据检验、数据库开发、Sitemesh集成、集成Hibernate/iBATIS

Struts2表现层框架

入门配置、核心组件、标签库、国际化、数据校验、Sitemesh集成转换器、拦截器、集成Hibernate/iBATIS

Spring业务层框架

入门配置、IoC容器、MVC、标签库、国际化、数据校验、数据库开发

Hibernate持久层框架

MySQL、Oracle、SQLServer iBATIS持久层框架

MySQL、Oracle、SQLServer

Web服务器（Tomcat/Jetty/Resin/JBossWeb）。

**Java系统架构师**

JavaWeb开源技术与框架

工作流

规则引擎

搜索引擎

缓存引擎

任务调度

身份认证

报表服务

系统测试

集群

负载平衡

故障转移

JavaWeb分布式开发技术

JTA（Java事物管理）

JAAS（Java验证和授权服务）

JNDI（Java命名和目录服务）

JavaMail（Java邮件服务）

JMS（java信息服务）

WebService(web服务)

JCA（java连接体系）

JMS（java管理体系）

应用服务器（JBossAS/WebLogic/WebSphere）。

JavaEE系统架构师

面向云架构（COA）

COA、SaaS、网格计算、集群计算、分布式计算、云计算

面向资源架构（ROA）

ROA、RESI

面向web服务架构（SOA）

WebService、SOA、SCA、ESB、OSGI、EAI

Java设计模式

创建式模式：抽象工厂/建造者/工厂方法/原型/单例

构造型模式：适配器/桥接/组合/装饰/外观/享元/代理

行为型模式：责任链/命令/解释器/迭代子/中介者/备忘录/观察者/状态/策略/模板方法/访问者

Java与UML建模

对象图、用例图、组件图、部署图、序列图、交互图、活动图、正向工程与逆向工程