

本文由qidongsoftware贡献

doc1。

### JAVA EE 架构师 需要具备的知识

1、构架师胚胎(程序员) 学习的知识是语言基础、设计基础、通信基础等，应该在大学完成，内容包括 java、c、c++、uml、RUP、XML、socket 通信(通信协议)——学习搭建应用系统所必须的原材料。 2、构架师萌芽(高级程序员) 学习分布式系统、组建等内容，可以在大学或第一年工作时间接触，包括 分布式系统原理、ejb、corba、com/com+、webservice(研究生可以研究网络计算机、高性能并发处理等内容) 3、构架师幼苗(设计师) 应该在掌握上述基础之上，结合实际项目经验，透彻领会应用设计模式，内容包括设计模式(c++版本、java 版本)、ejb 设计模式、J2EE 架构、UDDI、软件设计模式等。在此期间，最好能够了解软件工程在实际项目中的应用以及小组开发、团队管理。 4、软件构架师的正是成型在于机遇、个人努力和天赋软件构架师其实是一种职位，但一个程序员在充分掌握软构架师所需的基本技能后，如何得到这样的机会、如何利用所掌握的技能进行应用的合理构架、如何不断的抽象和归纳自己的构架模式、如何深入行业成为能够胜任分析、构架为一体的精英人才这可不是每个人都能够遇上的馅饼…… 第一阶段，think in java 关于 java 方面的特性都需要知道 第二阶段，要学会使用 jdk 的帮助文档，第三阶段，开始看 effective java 第四阶段，必看 java 模式 后面还需要了解 rup,uml 东西 这些完了，经过一些列项目经历下，2-4 年后你应该就可以到达 java 的中等水平了吧

Java 学习的 30 个目标以及系统架构师推荐的书 收藏 第一阶段 2. 你需要学习 JAVA 语言的基础知识以及它的核心类库 (collections, serialization, streams, networking, multithreading, reflection, event, handling, NIO, localization, 以及其他)。 5. 你需要学习 java 数据库技术，如 JDBC API 并且会使用至少一种 persistence/ORM 构架，例如 Hibernate，JDO，CocoBase，TopLink，InsideLiberator(国产 JDO 红工厂软件)或者 iBatis。 24. 你应该熟练掌握一种 JAVA IDE 例如 sunOne, netBeans, IntelliJIDEA 或者 Eclipse。(有些人更喜欢 VI 或 EMACS 来编写文件。随便你用什么了：) 26. 你需要熟悉一种单元测试体系(JUnit)，并且学习不同的生成、部署工具(Ant, Maven)。

27. 你需要熟悉一些在 JAVA 开发中经常用到的软件工程过程。例如 RUP(RationalUnifiedProcess)andAgilemethodologies。 第二阶段

1. 你需要精通面向对象分析与设计(0OA/0OD)、涉及模式(GOF, J2EEDP)以及综合模式。你应该十分了解 UML，尤其是 class, object, interaction 以及 statecharts。 3. 你应该了解 JVM, classloaders, classreflect，以及垃圾回收的基本工作机制等。你应该有能力反编译一个类文件并且明白一些基本的汇编指令。 6. 你还应该了解对象关系的阻抗失配的含义，以及它是如何影响业务对象的与关系型数据库的交互，和它的运行结果，还需要掌握不同的数据库产品运用，比如:oracle, mysql, mssqlserver。 7. 你需要学习 JAVA 的沙盒安全模式(classloaders, bytecodeverification, managers, policyandpermissions, codesigning, digitalsignatures, cryptography, certification, Kerberos，以及其他)还有不同的安全/认证 API，例如 JAAS(JavaAuthenticationandAuthorizationService), JCE (JavaCryptographyExtension), JSSE(JavaSecureSocketExtension)，以及 JGSS (JavaGeneralSecurityService)。 第三阶段 10. 你需要学习如何使用及管理 WEB 服务器，例如 tomcat, resin, Jrun，并且知道如何在其基础上扩展和维护 WEB 程序。 8. 你需要学习 Servlets, JSP，以及 JSTL(StandardTagLibraries)和可以选择的第三方 TagLibraries。 4. 如果你将要写客户端程序，你需要学习 WEB 的小应用程序(applet)，必需掌握 GUI 设计的思想和方法，以及桌面程序的 SWING, AWT, SWT。你还应该对 UI 部件的 JAVABEAN 组件模式有所了解。JAVABEANS 也被应用在 JSP 中以把业务逻辑从表现层中分离出来。(这条可有可无) 9. 你需要熟悉主流的网页框架，例如 JSF, Struts, Tapestry, Cocoon, WebWork，以及他们下面的涉及模式，如 MVC/MODEL2。 14. 你应该学习如何利用 JAVA API 和工具来构建 WebService。例如 JAXRPC(JavaAPIforXML/RPC) , SAAJ (SOAPwithAttachmentsAPIforJava) , JAXB(JavaArchitectureforXMLBinding) , JAXM(JavaAPIforX

MLMessaging) , JAXR(JavaAPIforXMLRegistries) , 或者 JWSDP(JavaWebServiceDeveloperPack)。 15. 你需要学习一门轻量级应用程序框架, 例如 Spring , PicoContainer , Avalon , 以及它们的 IoC/DI 风格(setter , constructor , interface injection)。 20. 你需要熟悉对不同有用的 API 和 frame work 等来为你服务。例如 Log4J(logging/tracing) , Quartz (scheduling) , JGroups(networkgroupcommunication) , JCache(distributedcaching) , Lucene(full-textsearch) , JakartaCommons 等等。

25. JAVA(精确的说是有些配置)是冗长的, 它需要很多的人工代码(例如 EJB) , 所以你需要熟悉代码生成工具, 例如 XDoclet 第四阶段 16. 你需要熟悉不同的 J2EE 技术 , 例如 JNDI(JavaNamingandDirectoryInterface) , JMS (JavaMessageService) , JTA/JTS(JavaTransactionAPI /JavaTransactionService) , JMX (JavaManagementExtensions) , 以及 JavaMail。 11. 你需要学习分布式对象以及远程 API , 例如 RMI 和 RMI/IIOP。 12. 你需要掌握各种流行中间件技术标准和与 java 结合实现 , 比如 Tuxedo、 CROBA , 当然也包括 javaEE 本身。 17. 你需要学习企业级 JavaBeans(EJB) 以及它们的不同组件模式 : Stateless/StatefulSessionBeans , EntityBeans( 包含 BeanManagedPersistence[BMP] 或者 Container-ManagedPersistence[CMP] 和它的 EJB-QL) , 或者 Message-DrivenBeans(MDB)。 13. 你需要学习最少一种的 XMLAPI , 例如 JAXP(JavaAPIforXMLProcessing) , JDOM(JavaforXMLDocumentObjectModel) , DOM4J , 或 JAXR(JavaAPIforXMLRegistries)。 18. 你需要学习如何管理与配置一个 J2EE 应用程序服务器 , 如 WebLogic , JBoss 等 , 并且利用它的附加服务 , 例如簇类 , 连接池以及分布式处理支援。你还需要了解如何在它上面封装和配置应用程序并且能够监控、调整它的性能。 第五阶段 (优先级低) 19. 你需要熟悉面向方面的程序设计以及面向属性的程序设计(这两个都被很容易混淆的缩写为 AOP) , 以及他们的主流 JAVA 规格和执行。例如 AspectJ 和 AspectWerkz。 21. 如果你将要对接或者正和旧的系统或者本地平台 , 你需要学习 JNI (JavaNativeInterface) and JCA (JavaConnectorArchitecture)。 22. 你需要熟悉 JINI 技术以及与它相关的分布式系统 , 比如掌握 CROBA。 23. 你需要 JavaCommunityProcess(JCP) 以及他的不同 JavaSpecificationRequests(JSRs) , 例如 Portlets(168) , JOLAP(69) , DataMiningAPI(73) , 等等。 28. 你需要能够深入了解加熟练操作和配置不同的操作系统 , 比如 GNU/linux , sunsolaris , macOS 等 , 做为跨平台软件的开发者。 29. 你还需要紧跟 java 发展的步伐 , 比如现在可以深入的学习 javaME , 以及各种 java 新规范 , 技术的运用 , 如新起的 web 富客户端技术。 30. 你必需要对 opensource 有所了解 , 因为至少 java 的很多技术直接是靠开源来驱动发展的 , 如 java3D 技术。 ==

===== 附 :  
JAVA 系统架构师应该看的几本书 Thinking in Java Effective Java UML 基础、案例与应用 UML 入门提高 软件工匠 设计模式——可复用面向对象软件的基础 重构-改善既有代码的设计 敏捷软件开发-原则、模式、实践 企业应用架构模式 Expert One-on-One J2EE Development without EJB 软件工程——实践者的研究方法 软件领导 - - 成功开发软件的指导准则 后面的两本书 , 其实已经有点属于项目经理的范畴了 , 不过还不是很深入 , 看看对做成功的系统架构师是很有好处。企业应用的系统架构师应该关注的几个方面 (具体情况具体分析, 以下未必准确 , 只是参考) 先来一些基础面试题 , 您答得出么 ? 1、说说 JVM 原理 ? 内存泄露与溢出区别 , 何时产生内存泄露 ? 2、用 java 怎么实现有每天有 1 亿条记录的 DB 存储 ? mysql 上亿记录数据量的数据库如何设计 ? 3、mysql 支持事务吗 ? DB 存储引擎有哪些 ? 4、mvc 原理 , mvc 模式的优缺点 , 如果让你设计你会怎么改造 MVC ? 5、hibernate 支持集群吗 ? 如何实现集群 ? 6、tomcat 最多支持并发多少用户 ? 7、map 原理 , 它是如何快速查找 key 的 ? map 与 set 区别 ? 8、描术算法 , 如何有效合并两个文件 : 一个是 1 亿条的用户基本信息 , 另一个是用户每天看电影连续剧等的记录 , 5000 万条。内存只有 1G ? ? ? 9、在 1 亿条用户记录里 , 如何快速查询统计出看了 5 个电影以上的用户 ? 可以参考 位图索引的原理 10、Spring 如何实现 IOC 与 AOP 的 , 说出实现原理 ?

数据持久层的设计 在 Spring 和 Hibernate , ibatis 出来以前，几乎每家公司都有自己的一套 方法和架构，而架构师的 50%的精力也会集中到这上面，EJB 只是增加架构师的 负担。在 Spring 出来以后，基本上，大多数的架构师都从重复设计这个轮子的 无用功中解脱出来了。Rod 的轮子太好用了，基本上，大家只要套上去就行了， 或者，剩下最重要的事情，是选择一个合适的数据库连接池的开源项目吧 MVC 架构的具体设计 MVC 只是个概要的概念，具体如何实现的具体技术很多，根据项目设计最恰当的架构 大并发性访问 太多了，包括从广域网到服务器到业务层再到架构再到程序、数据库的各种 调优，如使用缓存，静态化，静态动态 server 分离，在数据量达到一定程度时， 使用集群技术，优先考虑利用服务器的集群，其次是硬件集群，最后才是应用本 身加入集群功能 超大数据量返回结果 缓存命中率，分页，优化 SQL 语句，循环处理数据时尽可能共用对象，只保 留关键数据，及时释放内存占用 超大文件的读取和生成 尽可能快的读取大文件，并进行分析。写入大文件时，如何及时释放内存。 学会适当利用操作系统的命令行资源来更快完成任务。这方面经验比较少，以后 有空研究。 多线程的应用和管理 线程池的管理和监控，线程的启动（包括定时启动），结束，回收，线程资源的释放，这句话太简单了，待深入研究 用户界面可用性设计 平衡速度和可用性，恰当的使用异步和同步技术，展现关键数据为重点分布式的数据交流和集成 选择恰当的数据交互方式，从最泛滥低效的 Web Service 到最实用的文件共 享 群集系统的管理 如何确保缓存的同步？如何确保对象唯一性？如何保证各台机器的同步？ 是否采用 EJB?如何利用 J2EE 的特性（例如 JNDI）

复杂的业务规则 规则引擎和工作流引擎场景和应用 其实，作为一个真正的系统架构师，不应该局限于企业应用的系统，这种系 统往往有数据库的局限性，有时候，应该考虑是否可以横向跨越，直接对其它系 统做一些架构考虑，在没有丰富的实战经验的前提下，而只是看了其它人的系统 和代码，就能够给出有效的设计指导。 例如对于一个下载软件，可以有如下考虑： 1. 2. 3. 4. 5. 6. 未明和非法 url 的检验，已经下载失败的容许，信息记录 多线程下载一个文件，文件的切分和拼合，部分切片丢失的拼合可能性 下载线程管理 服务器或者 P2P 的机器之间的通讯协议 速度监控和限制 下载进度的监控和显示

作为一个在线播放软件，可以做如下考虑 1. 播放速度的保证 机器的问题基本不 存在了，关键是网络问题。如何在检测网络速度，根据影 片的质量，并缓冲足够多的 内容，保证播放一直尽可能顺利的完成。 2. 播放质量的保证 如何利用 DirectX 等 技术，最快的进行渲染，是自己写底层，还是利用已有的 API 由于没做过类似的项目， 可以写的东西还是少很多了。 系统架构师应该有的素质： 1、 实际的编程经验 最少 2 年吧，多了就不说了，其实从大学就开始钻研的话， 2、 书面表达能力和口头交 流能力 综合利用架构图， 图， UML 文字和代码片断， 表达自己设计思想， 至于是 Word 还是 ppt，应该通吃 在开发人员中发现架构师的最有价值标准是有效的沟通。 您需要技术娴熟、 经验丰富的开发人员， 这样的人员需要有就项目中的业务相关问 题进行沟通的经 历。架构师经常必须对理解方面的差距进行预计，然后才能有所贡献。 他们必须

愿意克服困难来确保技术和业务观点的融合。 他们并不必对意见交换工作进 行计 划和协调;这仍然主要是项目经理的工作。他们的任务是确定表述系统设计时的 最佳工具和构件，以促进有效的意见交换。他们必须能够判断当前方法显得不足 而需要 采用新方法的情况。写作技能也非常重要，还需要具有制作草图的技能或 使用制图软 件的能力。 3、 自觉主动;积极解决设计问题 架构师的日常工作目标经常并不明确。

很多开发人员直接参考功能规范来列 出任务清单。架构师通常则是向这些开发人员 提供所需结构的人员，以便尽可能 提高工作效率。好的候选者不仅进行沟通方面的 工作，而且也会预计各种设计问 题并加以解决——通常在没有任何具体指示的情况下自 觉进行。 无论所分配的职 责如何，积极参与项目的开发人员都有机会从一起工作的 人员中脱颖而出。 4、 抽象思维能力和总结能力 架构师，顾名思义，在系统未搭建 好之前，就要能够有一个草图在心。而如 果是对现有系统的改造，那么能在看过系 统的文档（如果有的话）和代码后，就 能总结出系统的架构特点。 架构师必须能够理 解表述模糊的概念并将其变成相关各方能够理解的项目 构件。他们必须能够理解抽象 概念，并以具体的语言对其进行沟通。开发人员中 好的候选者经常要求或自己主动解

释开发生命周期中容易混淆的问题。他们能迅速评估各种想法并将其纳入后续工作的操作建议中。开发人员经常具有很强的数学能力，而好的架构师则倾向于表现出更强的口头表达能力。管理人员经常说开发人员具有“工程意识”，而这是一个用于评估架构师的非常有意义的方面。架构师应该具有很强的解决技术问题的能力，但还必须能够准确获知更为全面的人员如何与技术交互的信息。这要求具有某种形式的抽象思维(而不再是代码的细节)，这种思维能力可能较难形成。5、全面的技术资讯吸收能力和选择鉴别能力 作为开发人员出身，对于某一个具体问题的研究能力(虽然很多人总结为 google 能力)，已经相当具备了。但是对技术资讯的全面接受和选择性深入了解能力，并且做出正确的判断，那些技术无非是厂家的噱头，而那些技术是真正可以应用到项目，提高项目质量的好技术，这种能力确实至关重要的。

本文来自 CSDN 博客，转载请注明出处：<http://blog.csdn.net/fish88168/archive/2010/09/10/5875944.aspx>

系统架构师知识结构 软件系统架构师综合的知识能力结构包括 9 个方面，即：

(1) 战略规划能力。 (2) 业务流程建模能力。 (3) 信息数据架构能力。 (4) 技术架构选择和实现能力。 (5) 应用系统架构的解决和实现能力。 (6) 基础 IT 知识及基础设施、资源调配的能力。 (7) 信息安全技术支持与管理保障能力。 (8) IT 审计、治理与基本需求分析、获取能力。 (9) 面向软件系统可靠性与系统生命周期的质量保障服务能力。作为系统架构师，必须成为所在开发团队的技术路线引导者；具有很强的系统思维的能力；需要从大量互相冲突的系统方法和工具中区分出哪些是有效的，哪些是无效的。架构师应当是一个成熟的、丰富的、有经验的、有良好教育的、学习快捷、善沟通和决策能力强的人。丰富是指他必须具有业务领域方面的工作知识，知识来源于经验或者教育。他必须广泛了解各种技术并精通一种特定技术，至少了解计算机通用技术以便确定哪种技术最优，或组织团队开展技术评估。优秀的架构师能考虑并评估所有可用来解决问题的总体技术方案。需要良好的书面和口头沟通技巧，一般通过可视化模型和小组讨论来沟通指导团队确保开发人员按照架构建造系统。因此，系统架构师知识维度可以总结为“多层次+多方面”。所谓多层次，意味着系统架构师必须在体系结构、计算机软硬件与网络基础知识、信息化基础知识、信息安全与可靠性基础知识等基本功的层面上受过良好的教育和快捷的学习能力；还须在系统架构设计方法、设计模式、设计流程以及各种模型等方面有丰富的经验，广泛了解各种构件产品和技术并精通一种特定领域的架构设计；进一步，还须在系统架构设计实践层面，有自己的认识和理解，同时具有很强的表述能力；所谓多方面，意味着系统架构师在每个知识层面上必须具有技术、管理、心理和艺术等多方面的知识和能力。这和系统架构师的多角色特点是相关的。本书也正是从这个角度来介绍系统架构的知识体系，即从系统构件、模式和规划三个方面

的技术基础、原理和方法的角度编写而成的关于软件架构师的基本知识结构和水平的教材。