**想要成为合格的Java程序员或工程师到底需要具备哪些专业技能，面试者在面试之前到底需要准备哪些东西呢？本文陈列的这些内容既可以作为个人简历中的内容，也可以作为面试的时候跟面试官聊的东西，你可以把这些内容写到你的简历中，当然更需要的是你在面试的时候向面试官展示这些专业技能。相信此文对正在寻觅Java程序员（Java工程师）职位的freshman以及希望成为中高级Java开发者的junior都会有所帮助。**

**专业技能**

1. **熟练的使用Java语言进行面向对象程序设计，有良好的编程习惯，熟悉常用的Java API，包括集合框架、多线程（并发编程）、I/O（NIO）、Socket、JDBC、XML、反射等。[泛型]**
2. **熟悉基于JSP和Servlet的Java Web开发，对Servlet和JSP的工作原理和生命周期有深入了解，熟练的使用JSTL和EL编写无脚本动态页面，有使用监听器、过滤器等Web组件以及MVC架构模式进行Java Web项目开发的经验。**
3. **对Spring的IoC容器和AOP原理有深入了解，熟练的运用Spring框架管理各种Web组件及其依赖关系，熟练的使用Spring进行事务、日志、安全性等的管理，有使用Spring MVC作为表示层技术以及使用Spring提供的持久化支持进行Web项目开发的经验，熟悉Spring对其他框架的整合。**
4. **熟练的使用Hibernate、MyBatis等ORM框架，熟悉Hibernate和MyBatis的核心API，对Hibernate的关联映射、继承映射、组件映射、缓存机制、事务管理以及性能调优等有深入的理解。**
5. **熟练的使用HTML、CSS和JavaScript进行Web前端开发，熟悉jQuery和Bootstrap，对Ajax技术在Web项目中的应用有深入理解，有使用前端MVC框架（AngularJS）和JavaScript模板引擎（HandleBars）进行项目开发的经验。**
6. **熟悉常用的关系型数据库产品（MySQL、Oracle），熟练的使用SQL和PL/SQL进行数据库编程。**
7. **熟悉面向对象的设计原则，对GoF设计模式和企业应用架构模式有深入的了解和实际开发的相关经验，熟练的使用UML进行面向对象的分析和设计，有TDD（测试驱动开发）和DDD（领域驱动设计）的经验。**
8. **熟悉Apache、NginX、Tomcat、WildFly、Weblogic等Web服务器和应用服务器的使用，熟悉多种服务器整合、集群和负载均衡的配置。**
9. **熟练的使用产品原型工具Axure，熟练的使用设计建模工具PowerDesigner和Enterprise Architect，熟练的使用Java开发环境Eclipse和IntelliJ，熟练的使用前端开发环境WebStorm，熟练的使用软件版本控制工具SVN和Git，熟练的使用项目构建和管理工具Maven和Gradle。**

**说明：上面罗列的这些东西并不是每一项你都要烂熟于心，根据企业招聘的具体要求可以做相应的有针对性的准备。我个人觉得前6项应该是最低要求，是作为一个Java开发者必须要具备的专业技能。**

**项目经验**

**项目介绍**

**本系统是X委托Y开发的用于Z的系统，系统包括A、B、C、D等模块。系统使用了Java企业级开发的开源框架E以及前端技术F。表示层运用了G架构，使用H作为视图I作为控制器并实现了REST风格的请求；业务逻辑层运用了J模式，并通过K实现事务、日志和安全性等功能，通过L实现缓存服务；持久层使用了M封装CRUD操作，底层使用N实现数据存取。整个项目采用了P开发模型。**

**说明：上面的描述中，E通常指Spring（Java企业级开发的一站式选择）；F最有可能是jQuery库及其插件或者是Bootstrap框架，当然如果要构建单页应用（SPA）最佳的方案是前端MVC框架（如AngularJS）和JavaScript模板引擎（如HandleBars）；G显然是MVC（模型-视图-控制），最有可能的实现框架是Spring MVC，除此之外还有Struts 2、JSF以及Apache为JSF提供的MyFaces实现，可以使用JSP作为MVC中的V，也可使用模板引擎（如Freemarker和Velocity）来生成视图，还可以是各种文档或报表（如Excel和PDF等），而Servlet和自定义的控制器是MVC中的C，当然Spring MVC中提供了作为前端控制器的DispatcherServlet；J通常是事务脚本，K应该是AOP（面向切面编程）技术，L目前广泛使用的有memcached和Redis；M的选择方案很多，最有可能的是Hibernate和MyBatis，也可以两种技术同时运用，但通常是将增删改交给Hibernate来处理，而复杂的查询则由MyBatis完成，此外TopLink、jOOQ也是优秀的持久层解决方案；底层的数据存取传统上是使用关系型数据库，可以是MySQL、Oracle、SQLServer、DB2等，随着大数据时代的来临，也可以采用NoSQL（如MongoDB、MemBase、BigTable等）和其他大数据存取方案（如GFS、HDFS等）；项目的开发模型P可以是瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、RAD模型等。**

**项目开发流程**

1. **可行性分析 >>> 可行性分析报告 / 项目开发计划书**
2. **需求分析 >>> 需求规格说明书** 
   * **OOAD（用例图、时序图、活动图）**
   * **界面原型：帮助理解需求、业务层设计时推导事务脚本**
3. **设计 >>> 概要设计说明书/详细设计说明书** 
   * **抽取业务实体（领域对象）：类图、E-R图（概念设计阶段）**
   * **分层架构：确定各层的技术实现方案（具体到使用的框架、数据库服务器、应用服务器等）。业务层设计：事务脚本模式（事务：用户发送一次请求就是一个事务；脚本：一个方法或一个函数；事务脚本：把一次请求封装为一个方法或一个函数；事务脚本模式：一个事务开始于脚本的打开，终止于脚本的关闭）。业务层涉及的对象有三种类型：事务脚本类（封装了业务的流程）、数据访问对象（DAO，封装了持久化操作）、数据传输对象（DTO，封装了失血/贫血领域对象），三者之间的关系是事务脚本类组合（聚合）数据访问对象，这二者都依赖了数据传输对象**
   * **正向工程（UML类图生成Java代码）和逆向工程（Java代码生成UML类图）**
   * **数据库物理设计（ER图转换成表间关系图、建库和建表、使用工具插入测试数据）**
4. **编码**
5. **测试 >>> 测试报告 / 缺陷报告** 
   * **单元测试：对软件中的最小可测试单元进行检查和验证，在Java中是对类中的方法进行测试，可以使用JUnit工具来实施。**
   * **集成测试：集成测试也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求组装成为子系统进行测试。**
   * **系统测试：将已经确认的软件、硬件、外设、网络等元素结合在一起，进行信息系统的各种组装测试和确认测试，系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。**
   * **验收测试：在软件产品完成了单元测试、集成测试和系统测试之后，产品发布之前所进行的软件测试活动。它是技术测试的最后一个阶段，也称为交付测试。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。**
6. **交付和维护 >>> 用户手册 / 操作手册**

**项目管理**

* **版本控制：CVS/SVN/Git**
* **自动构建：Ant/Maven/Ivy/Gradle**
* **持续集成：Hudson/Jenkins**

**系统架构**

* **负载均衡服务器：F5、A10**
* **应用服务器：** 
  + **HTTP服务器：Apache、NginX（HTTP、反向代理、邮件代理服务器）**
  + **Servlet容器：Tomcat、Resin**
  + **EJB容器：WildFly（JBoss Application Server）、GlassFish、Weblogic、Websphere**
* **数据库服务器：MySQL、Oracle**

**第三方工具（插件）应用**

* **图表工具：基于jQuery的图表插件（如jQchart、Flot、Charted等）、Chart.js、Highcharts等。[fusioncharts]**
* **报表工具：Pentaho Reporting、iReport、DynamicReports等。**
* **文档处理：POI、iText等。**
* **工作流引擎：jBPM、OpenWFE、Snaker、SWAMP等。**
* **作业调度：Quartz、JobServer、Oddjob等。**
* **缓存服务：EhCache、memcached、SwarmCache等。**
* **消息队列：Open-MQ、ZeroMQ等。**
* **安全框架：Shiro、PicketBox等。**
* **搜索引擎：IndexTank、Lucene、ElasticSearch等。**
* **Ajax框架：jQuery、ExtJS、DWR等。**
* **UI插件：EasyUI、MiniUI等。**
* **富文本框：UEditor、CKEditor等。**

**面试提问**

* **项目是为哪个公司开发的？项目的投入是多少？**
* **有多少人参与了项目开发？整个团队中，测试人员、开发人员、项目经理比例是多少？**
* **项目开发了多长时间？项目总的代码量有多少？你的代码量有多少？**
* **项目采用了怎样的开发模型或开发流程？项目的架构是怎样的？项目的技术选型是怎样的？**
* **你在项目中承担了怎样的职责？是否经常开会或加班？项目完成后有哪些收获或是经验教训？**
* **项目中最困难的部分是什么？如何解决团队开发时遇到的各种冲突？**

**说明：对于没有实际项目经验的，可以在前程无忧、智联招聘、拉勾网等网站上搜索招聘Java程序员的公司，找到他们的官方网站了解他们做的项目，查看项目的详细介绍，然后尝试完成其中一部分功能，最好请教一下高人看看自己的设计和代码是否恰当，这样相当于积累了一定的项目经验。**

**面试题**

**Java常见的面试题已经总结成《Java面试题集》、《Java面试题全集》以及《面试编程题拾遗》等文章陆续发布在我的CSDN博客，各大公司的面试题我会继续整理发布。**

**其他**

**常见错误**

1. **只在计算机上练习**
2. **不做行为面试题演练**
3. **不做模拟面试训练**
4. **试图死记硬背答案**
5. **不大声说出你的解题思路**
6. **代码不够严谨**
7. **不写测试代码**
8. **轻言放弃**

**面试时可以反问面试官的问题**

1. **我注意到你们使用了X技术，请问你们是如何解决Y问题的？**
2. **为什么你们的产品使用了X技术而不是Y技术？据我所知，X技术虽然有A、B、C等好处，但也存在D和E问题，而Y技术可以解决D和E问题。**
3. **我对您说的X技术不是太熟悉，但我感觉它是一个不错的解决方案，您能多讲讲它的工作原理吗？**
4. **你们团队是如何进行项目规划的？一周会有几次例会？每周的代码量大概是多少？**
5. **就X问题我能想到的解决方案目前就只有Y了，请问您会怎么解决这个问题？**

**S.A.R.法则**

**S.A.R法则是指先描述问题的场景，然后解释你采取的行动，最后陈述结果。**

**算法题的五种解法**

1. **举例法：通过举例子发现其中的一般规则。**

**例子：圆内接三角形是锐角三角形的概率是多少？这是搜狗的一个面试题，可以在圆上随意画三个点连接成三角形就可以知道答案了。**

1. **模式匹配法**

**例子：一个有序数组的元素经过循环移动，元素的顺序变成"3 4 5 6 1 2"。怎样找到数组中最小的那个元素，假设数组中的元素各不相同。这个题目和折半查找看起来是那么相似，因此可以借鉴折半查找的算法，最小元素显然出现在"mid > right"的转折点。**

1. **简化推广法**

**说明：简化问题规模和数据类型，然后再发现通用的解法。**

1. **简单构造法**

**例子：找出"abcde"的所有可能的排列组合。先考虑只有"a"的情况，再考虑"ab"的情况，以此类推。最终你可能会得到一个递归公式。这种方法往往会演变成递归法。**

1. **数据结构头脑风暴法**

**例子：随机生成一些数字，并找出其中位数。这种问题可以在头脑中将你了解的数据结构过一遍，看看哪种是最合适的数据结构。上面的题目可以建立两个堆，一个大根堆和一个小根堆，较小的元素放在大根堆，较大的元素放在小根堆，如果两个堆不平衡，可以从其中一个堆取出元素放入另一个堆即可。最后中位数应该是两个堆的根之一。**

**录用谈判**

1. **要理直气壮的提出具体的待遇要求**
2. **开出比预期稍高的价码**
3. **不要只盯着薪水（很多公司更愿意就薪水之外的条件做出让步）**
4. **使用最合适的方法（可以尝试在电话或E-mail中谈判待遇）**

**自我评价**

1. **学习能力（搞IT行业的人需要不断的了解新的技术、工具和方法）**
2. **团队意识（相比个人英雄主义，IT行业更倡导团队合作）**
3. **抗压能力（很多IT企业的工作强度相对来说还是比较大的）**