

Java学习路径（三）过程篇 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_Java_E5_AD_A6_E4_B9_A0_c97_253046.htm 第二步是学习Java的语法。Java的语法是类C的，基本上主流的编程语言不是类C，就是类C的，没有什么新东西，所以语法的学习，大概就是半天的时间足够了。唯一需要注意的是有几个不容易搞清楚的关键字的用法，public，protected，private，static，什么时候用，为什么要用，怎么用，这可能需要有人来指点一下，我当初是完全自己琢磨出来的，花了很长的时间。不过后来我看到

《Thinking in Java》这本书上面是讲了这些概念的。第三步是学习Java的面向对象的编程语言的特性的地方。比如继承，构造器，抽象类，接口，方法的多态，重载，覆盖，Java的异常处理机制。对于一个没有面向对象语言背景的人来说，我觉得这个过程需要花很长很长时间，因为学习Java之前没有C的经验，只有C的经验，我是大概花了一个月左右吧，才彻底把这些概念都搞清楚，把书上面的例子反复的揣摩，修改，尝试，把那几章内容反复的看过来，看过去，看了不下5遍，才彻底领悟了。不过我想如果有C经验的话，应该一两天时间足够了。那么在这个过程中，可以多看看《Thinking in Java》这本书，对面向对象的讲解非常透彻。可惜的是我学习的时候，并没有看到这本书，所以自己花了大量的时间，通过自己的尝试和揣摩来学会的。第四步就是开始熟悉Java的类库。Java的基础类库其实就是JDK安装目录下面jre\lib\rt.jar这个包。学习基础类库就是学习rt.jar。基础类库里面的类非常多。据说有3000多个，我没有统计过。但是真正对于我

们来说最核心的只有4个，分别是 `Java.lang.*`、`Java.io.*`、`Java.util.*`、`Java.sql.*`。这四个包的学习，每个包的学习都可以写成一本厚厚的教材，而Oreilly也确实是这样做的。我觉得如果时间比较紧，是不可能通过读四本书来学习。我觉得比较好的学习方法是这样的：首先要通读整个package的框架，了解整个package的class，interface，exception的构成，最好是能够找到介绍整个包框架的文章。这些专门介绍包的书籍的前几章应该就是这些总体的框架内容介绍。对包整体框架的把握并不是要熟悉每个类的用法，记住它有哪些属性，方法。想记也记不住的。而是要知道包有哪些方面的类构成的，这些类的用途是什么，最核心的几个类分别是完成什么功能的。我在给人培训的时候一般是一次课讲一个包，所以不可能详细的介绍每个类的用法，但是我反复强调，我给你们讲这些包的不是要告诉你们类的方法是怎么调用的，也不要求你们记住类的方法调用，而是要你们了解，Java给我们提供了哪些类，每个类是用在什么场合，当我遇到问题的时候，我知道哪个类，或者哪几个类的组合可以解决我的问题，That all！，当我们具体写程序的时候，只要你知道该用哪个类来完成你的工作就足够了。编码的时候，具体的方法调用，是边写代码，边查Documentation，所有的东西都在Documentation里面，不要求你一定记住，实际你也记不住3000多个类的总共将近10万个方法调用。所以对每个包的总体框架的把握就变得极为重要。第五步，通过上面的学习，如果学的比较扎实的话，就打好了Java的基础了，剩下要做的工作是扫清Documentation里面除了上面4个包之外的其他一些比较有用处的类。相信进展到这一步，Java的自学能力已经被培养出

来了，可以到了直接学习Documentation的水平了。除了要做GUI编程之外，JDK里面其他会有用处的包是这些：
Java.text.*. Java.net.*. Javax.naming.*. 这些包里面真正用的比较多的类其实很少，只有几个，所以不需要花很多时间。第六步，Java Web 编程 Web编程的核心是HTTP协议，HTTP协议和Java无关，如果不熟悉HTTP协议的话，虽然也可以学好Servlet/JSP编程，但是达不到举一反三，一通百通的境界。所以HTTP协议的学习是必备的。如果熟悉了HTTP协议的话，又有了Java编程的良好基础，学习Servlet/JSP简直易如反掌，我学习Servlet/JSP就用了不到一周的时间，然后就开始用JSP来做项目了。在Servlet/JSP的学习中，重头仍然是Servlet Documentation。Servlet API最常用的类很少，花比较少的时间就可以掌握了。把这些类都看一遍，多写几个例子试试。Servlet/JSP编程本质就是在反复调用这些类来通过HTTP协议在Web Server和Browser之间交谈。另外对JSP，还需要熟悉几个常用JSP的标记，具体的写法记不住的话，临时查就是了。此外Java Web编程学习的重点要放在Web Application的设计模式上，如何进行业务逻辑的分析，并且进行合理的设计，按照MVC设计模式的要求，运用Servlet和JSP分别完成不同的逻辑层，掌握如何在Servlet和JSP之间进行流程的控制和数据的共享，以及Web Application应该如何配置和部署。第七步，J2EE编程 以上的学习过程如果是比较顺利的话，进行到这一步，难度又陡然提高。因为上面的知识内容都是只涉及一个方面，而像EJB，JMS，JTA等核心的J2EE规范往往是几种Java技术的综合运用的结晶，所以掌握起来难度比较大。首先一定要学习好JNDI，JNDI是App Server定位服务器资源

(EJB组件, Datasource, JMS) 查找方法, 如果对JNDI不熟悉的话, EJB, JMS这些东西几乎学不下去。JNDI其实就是`javax.naming.*`这个包, 运用起来很简单。难点在于服务器资源文件的配置。对于服务器资源文件的配置, 就需要看看专门的文档规范了, 比如`web.xml`的写法, `ejb-jar.XML`的写法等等。针对每种不同的App Server, 还有自己的服务资源配置文件, 也是需要熟悉的。然后可以学习JTA, 主要是要理解JTA对于事务的控制的方法, 以及该在什么场合使用JTA。这里可以简单的举个例子, 我们知道一般情况可以对于一个数据库连接进行事务控制(`conn.setAutoCommit(false), ..., conn.commit()`), 做为一个原子操作, 但是假设我的业务需求是要把对两个不同数据库的操作做为一个原子操作, 你能做的到吗? 这时候只能用JTA了。假设操作过程是先往A数据库插一条记录, 然后删除B数据库另一个记录, 我们自己写代码是控制不了把整个操作做为一个原子操作的。用JTA的话, 由App Server来完成控制。在学习EJB之前要学习对象序列化和RMI, RMI是EJB的基础。接着学习JMS和EJB, 对于EJB来说, 最关键是要理解EJB是如何通过RMI来实现对远端对象的调用的, 以及在什么情况下要用到EJB。在学习完EJB, JMS这些东西之后, 你可能会意识到要急不可待学习两个领域的知识, 一个是UML, 另一个是Design Pattern。Java企业软件的设计非常重视框架(Framework)的设计, 一个好的软件框架是软件开发成功的必要条件。在这个时候, 应该开始把学习的重点放在设计模式和框架的学习上, 通过学习和实际的编程经验来掌握EJB的设计模式和J2EE的核心模式。J2EE规范里面, 除了EJB, JMS

，JTA，Servlet/JSP，JDBC之外还有很多很多的企业技术，这里不一一进行介绍了。另外还有一个最新领域Web Services。Web Services也完全没有任何新东西，它像是一种黏合剂，可以把不同的服务统一起来提供一个统一的调用接口，作为使用者来说，我只要获得服务提供者给我的WSDL（对服务的描述），就够了，我完全不知道服务器提供者提供的服务究竟是EJB组件，还是.Net组件，还是什么CORBA组件，还是其他的什么实现，我也不需要知道。Web Services最伟大的地方就在于通过统一的服务提供方式和调用方式，实现了整个Internet服务的共享，是一个非常令人激动的技术领域。Web Services好像目前还没有什么很好的书籍，但是可以通过在网络上面查资料的方式来学习。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com