文档来自新浪微博 @软件教书匠金旭亮

“计算机可以改变世界”，这种观点倒推三十年，是没有几个人同意的，直到本世纪初互联网兴起并走入寻常百姓家庭之后，就再也没有人怀疑这件事了。近些年来，以互联网为代表的IT行业经历了“爆炸性”的增长，颠覆或重塑了“一堆的”传统产业，IT业的急速扩张，导致对人才极度饥渴，就象是黑洞一样，再多的人也不够填的。

在这个时代背景下，跨专业报考计算机专业研究生，跨行跳槽当程序员，成为两个全社会级别的现象，时人戏称：“条条大路通CS”。

这世界上再没有比商人对商机的嗅觉更灵敏的了，既然有那么多的人想当程序员，这个庞大的现实的需求就需要被满足，于是程序员就业培训就成了一个完整的产业，出于对利润的追逐，培训机构普遍给与学员这样的承诺：“到我这学X个月，包你找到每月XXK的工作”，“15周培训出一个全栈程序员”，……，培训机构还采用了灵活的商业策略，比如“找不到工作，免费再学”、“免费先学，毕业后以工资偿还培训费”、“打通学校关节，将参加培训作为学生毕业的必修学分”、“挂招聘的羊头，卖培训的狗肉”……，可谓“八仙过海，各显神通”。

这很正常。

商业运作嘛，宣传不夸张一点，那还叫广告吗？不用点营销策略，那还是商业吗？这些其实都无可厚非，只要不违法就好。

我关注的一点是——***程序员是可以“速成”的吗？如果可以，那怎样“速成”？***

就我个人的观点，单纯地回答说“可以”还是“不可以”，都是片面的。

如果以培养“搬砖”程序员为目标，通过几个月高强度的专门训练，“速成”确实是可以做得到的，并且在整个IT行业规模迅速扩张的大背景下，这些“仅能搬砖”的程序员，同样可以得到比他在原专业原工作岗位收入更高的工作。

至于说中级以上程序员也可以“速成”，则完全是胡扯了。

这里我们需要知道，***“速成”的程序员能顺利找到工作的大前提是“整个IT行业规模持续地迅速扩张”***， 我在我的知乎Live——[如何自学编程](https://www.zhihu.com/lives/860490807847841792" \t "http://weibo.com/ttarticle/p/_blank)，中谈到过“**技术人才市场规律**”的作用，这里复述一下：

IT行业当前较高的平均收入的诱惑，会让很多人想转行干程序员，其结果是短时间内就会有大量的人涌入这个行当，特定领域的人才需求会迅速地由***短缺***转向***供需平衡***再到 ***供过于求***。

在市场的不同阶段，用人单位的招聘策略也是不一样的。

当供求平衡或供过于求时，用人单位在人才招聘这块必然会对求职者所必须具备的职业技能提出越来越高的要求，那时，速成的“搬砖”程序员好找工作的日子也就到头了，就算现在暂时己经占了位置的，如果不能及时补上自己专业技能的短板，在越来越充足的人力供给下，被更有竞争力的“后来者”把饭碗抢走，也不会是稀罕事。

其实这种情形一点也不新鲜，离我们比较近的一次发生在“Android/iOS手机开发”领域，现在我看己经轮到了“Web前端”这个领域了……

不再谈“市场规律”了，让我们把目光移向“速成”这件事上。我前面说过“搬砖程序员”的“速成”是可能的，想成为这样的程序员，虽然我承认上培训班是一条现实的可行的方式，但我更推崇通过自学达到这个目标，并认为只要方法得当并且足够勤奋，自学并不比上培训班慢多少，而且自学往往更能锻炼出真才实学。

​ 有关“自学还是上培训班”，两者利弊对比，我在另一篇专栏文章——[《自学编程 vs. 参加培训》](http://weibo.com/ttarticle/p/show?id=2309404126098286892132&mod=zwenzhang" \t "http://weibo.com/ttarticle/p/_blank)中己经有了集中的充分的论述，这里我就不重复了。下面就谈谈一名自学者如何通过提升自学的效率，从“菜鸟”“速成”为初级程序员。

其实要“速成”方法并不复杂，其中关键有以下几点：

***（1）“只学必要的知识，不做无用功”***，

***（2）“充分收集信息，少走弯路”，***

***（3）“遵循科学的刻意训练方法，循序渐进”***

在本文中，我只针对第1点展开，后面两点以后有机会再写文章展开介绍。

第1点的另外一种叙述方式，或者说具体行动指南是：

***集中你的时间与精力，在短时间内掌握完成特定任务所需的“最少必要”的知识和技能。***

虽然软件技术本身极其庞大，里头有N多个领域，知识总量之多无庸多说，但从“学以致用”的角度来看，针对特定的应用场景，掌握特定技术，并达到能使用它来干活的水平，要学的东西其实并不多。

只要你针对用人单位急需人才的那个技术领域，有针对性地依据用人单位的具体应用场景，去学习和掌握相应的专业技能，那么，你并不需要学N多东西，也能得到相应的工作机会。

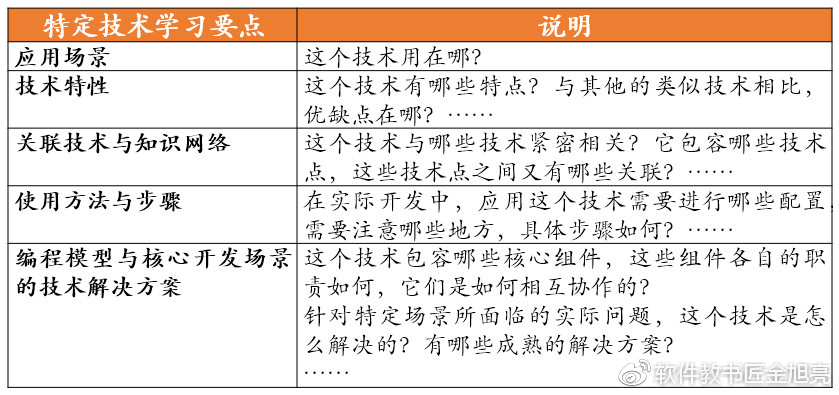
这就是达到“速成”的关键——***选准目标，一剑封喉！***

这就要求你全面地收集用人单位的信息，知彼知己，依据自己的具体情况，做出学习什么技术的决策。

这件事，自己做，别人帮不上忙。

我能帮的，是帮助你理清思路，找到“高效学习软件技术”的方法。

为此，我拟定了一个的“特定领域技术自学表”，将自学掌握特定软件技术所要做的工作总结为五项，如下所示：

软件技术自学模板

​自学的过程，就是填表的过程。

为了更直观形象地说明，下面我就以学习与掌握Java技术领域中的NIO这块技术为例来介绍，展示这个表所填的具体内容。

## 第1项：明确技术的具体应用场景

NIO这个技术，主要用于开发高性能的网络服务端应用，这样的网络应用，具有应对高并发请求的能力。

## 第2项：了解技术特性

NIO这项技术：

（1）它是从“缓冲区（Buffer）”而不是“流（Stream）”中读取数据。

（2）它采用I/O非阻塞的操作。

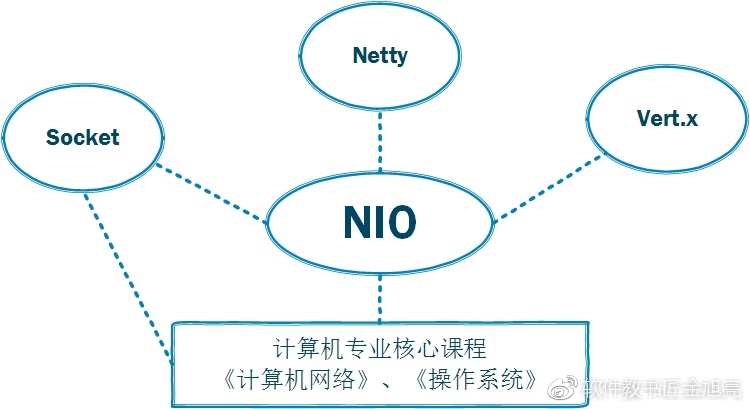
（3）它使用单Selecor多Channel的方式，让一个线程管理多个网络连接，从而避免了单线程单连接的编程方式在高并发请求环境下所带来的问题：Server端大量地创建线程，占用大量服务器资源。

在了解特定技术的技术特性时，往往会涉及许多的术语和概念，把这些术语和概念“打磨”清楚，并能用自己的话把它讲出来，是衡量你是否真弄明白了这个技术的“试金石”。

没有真正地理解特定技术领域中的核心术语与相关概念，是很多人学技术学得稀里胡涂的原因所在。

## 第3项：关联技术与知识网络

通过阅读相关的书籍，在互联网上搜索相关的文章，不难绘出以下的技术关联图：

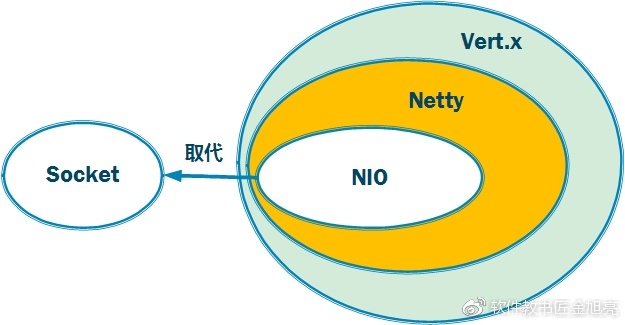
与NIO技术相关联的其他技术

​

可以看到，NIO与经典Socket开发，以及两个流行的高性能网络开发框架Netty和Vert.x密切相关。

另外，掌握上述技术，必须具备计算机专业的两门核心课程《计算机网络》和《操作系统》中的部分知识，比如Socket编程就直接与操作系统的网络组件及进程和线程管理机制密切相关。

我们还可以进一步地明确这些技术之间关联关系到底是什么？

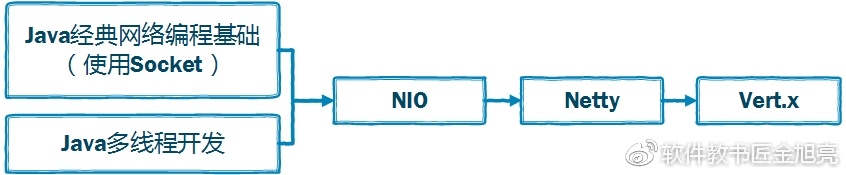
各技术之间的依赖与关联关系

​

对着这个图，自己就说了：

*在需要高性能的场景，应该使用NIO取代传统的单线程Socket，如果想提高开发效率，避免被原生的Java NIO各种技术细节所困扰，可以使用更上一层封装的Netty或Vert.x开源框架 ，其中Netty封装了NIO，而Vert.x又是基于Netty而构建的。*

下面就可以进一步地确定出学习路线图：

拟定的NIO技术学习路线

​

明确关联技术和学习路线，对于自学者来说非常关键，对于真正靠谱的培训班来说，它通常已经制订好了一条适合于多数学员的学习路线，从这个意义上来说，上培训班节约了时间，因为你无需再花时间自己总结了。

不那么靠谱的培训班，从节约时间角度出发，可能会直接选择从最上层的框架（上图就是Vert.x或Netty）切入，类似于中学高考时针对特定的题型进行专门训练，培训班选择几个典型的应用场景，设计一些典型的项目，教给学员固定的编程套路，让他们“照猫画虎”，“用熟背熟”，只要能通过用人单位的面试，这就OK了，至于框架内部的运作机理，封装的是什么，又是如何封装的，一概不管不教，因为根本不可能有足够的时间让学员去慢慢地消化吸收和理解这些知识，而且有不少学员也没有具备相应的基础，打好专业理论与技术基础，不是就业培训班应该承担的任务。

这种培训方式，学员就只学了一点皮毛，只知其然不知其所以然，就算能通过面试找到工作，当他在实际工作中面临到一个以前培训班没介绍过的变化了的场景时，多半就“草鸡”了。

所以我说，上培训班“吃现成饭”固然节省了时间，但同时也剥夺了你锻炼和提升自己能力的机会。

为什么这么说？

因为自己***学会从互联网上海量的信息中搜索、定位所需的信息，并对这些信息进行判断、分析、综合的能力，对于解决实际工作中遇到的各种问题，对于日后成长为一名优秀的程序员来说，非常关键***。这种能力，培训班是无法教给你的，只能靠你自己主动进行“刻意训练”。

## 第4项：这个技术的使用方法

掌握一个技术的最快捷方式，就是走一个HelloWorld流程。

***所谓 “ HelloWorld 流程 ”，就是使用某项技术写程序，走完从编码到部署的全过程。***

NIO由于是内置于JDK中的，直接可用，因此不够典型，下面就以其上层框架Netty为例进行介绍。

在学习Netty初期要完成的Hello World流程大致是这样的：

（1）打开你熟悉的IDE（Eclipse/Netbeans/IntelliJ等等），创建一个Maven项目

（2）在pom.xml中添加相关依赖，导入Netty相关的jar包

（3）照着书或网络上的例子，写一个最简单的Netty应用

（4）将你的Netty应用打包，然后部署到Linux服务器上。

将上面四步走完，你就基本上掌握了这个技术的使用方法。

在具体学习某项技术时，其官网上通常都会有一个相应的HelloWorld教程，跟着做就是了。虽然官网教程大多是英文的，但国内有不少热爱学习的技术发烧友做了很无私的贡献，积极地将这些教程译为中文，直接读这些中文教程，你完成一个HelloWorld流程可能花不了一个小时。

另外，国内也有很多热爱学习的人将他们探索新技术、走完HelloWorld流程的过程写成了博客，百度一下就能找到这些文章，这些文章通常都很接地气，不仅有详细的步骤和丰富的截图，而且可能会掉入的各种坑都告诉你了，跟着做，难度很低。

这一步其实花时间不多，但效果很好，实乃“速成”绝招。

不过这类走HelloWorld流程的文章仅能起到一个快速了解和入门的作用，要想真正地深入，还是得看专门的书籍，或者到官网上看技术文档和教程，深入学习、主动实践，才能真正掌握好这个技术。

## 第5项：编程模型与核心开发场景的技术解决方案

这一块是学习技术最花费时间的地方。只有完成了这块，你才能说真正掌握了这项技术。

任何一项技术，都有其编程模型，对于面向对象的软件开发框架来说，最关键的就是它所包容的核心组件（或类型）。

还是拿NIO来说事，NIO其实就是三大组件：

NIO的三大核心组件

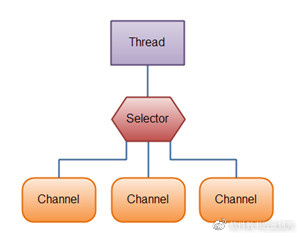
​

了解这三大组件的职责，掌握它们的用法，就是掌握NIO这个技术领域“最少的最必要”的知识与技能。

先提溜出Channel与Buffer两个，通过学习相应资料，按照着自己的理解画一张图：

Channel与Buffer是如何协作的

​ Selector的用法，总结成另一张图：

Selector与Channel如何协作？

​

在自学过程中通过动手画图这种方式，**将知识进行可视化**，你就能更好地理解与把握知识。

这些图，有些在网上可以找到，有些则需要你自己画，只要你能对着这些图说清楚：

（1）每个组件的职责是什么？

（2）这些组件之间是如何协作的？

你就基本上把握了这个技术领域的最关键也最必要的知识。

剩余的就是写Demo了，写一堆的小的Demo，每个Demo针对一个小的场景，测试一个特定的技术特性。

小的Demo写了一堆之后，通常你就会比较熟悉这个技术了，这时，你再“来一个大招”，基于你所掌握的这个技术写一个能干点事的有用的规模比较大的项目，比如，用学会的NIO技术写一个网络文件服务器，你可以把这个项目放到GitHub上，求职时，直接给出GitHub上的链接，这可比你费心思去“包装”简历、伪造项目经验有效得多……

好了，现在再回过头来“点题”：

***初级程序员可以“速成”吗？可以不上培训班吗？***

当然可以！

你需要依据用人单位的具体需求，明确胜任特定岗位所需的“最小必要”的知识与技能集合，列出相关的技术领域，每个技术领域按照我上面给出的表格模板进行“填充”，就能在一个比较短的时间内具备相应的工作技能，为你得到这个工作机会铺平道路。

上述自学过程中，唯一可能存在问题的地方就是——***你如何确定在特定技术领域内应付特定工作任务所需的“最小必要”的知识与技能集合包容些什么？***

这个确实是个问题，但也不是不能解决的。

最根本的解决方法是采用“***真实项目驱动***”的方式来确定，通常是自己做一个模拟项目，或者是收集相关信息分析一个此领域内的真实项目，就知道该学什么了，另外，直接询问有经验的人，也是一个很有效的方法。

=========================================

说明：

本文是“[如何自学计算机专业课程](https://www.zhihu.com/lives/837669764146003968" \t "http://weibo.com/ttarticle/p/_blank)”，“[如何自学编程](https://www.zhihu.com/lives/860490807847841792" \t "http://weibo.com/ttarticle/p/_blank)”知乎两场Live之补充学习资料。 因为Live时间有限，有不少东西没有能展开，所以以后会陆续地发一些微博和文章，作为Live所讲授观点与方法的实践指南和补充。

另外，我将于10月11日针对Java自学者开讲第三场Live——“[如何自学Java：入门与编程技能训练](https://www.zhihu.com/lives/889495940065538048" \t "http://weibo.com/ttarticle/p/_blank)”，针对初学者介绍Java技术领域的“最小必要知识与技能”集合，并给出相应的学习路线和学习建议，帮助自学者节省时间，少走弯路。

最后说一句：***学会自学，是在这个时代生存和发展的前提。***

希望我的工作和努力，能帮助大家掌握自学的方法！