

Java环境变量，真的还有必要配吗？ - 陈本布衣 - 博客园

陈本布衣关注 - 27 粉丝 - 109 + 加关注

作为年龄上堪称老鸟而技术上却是菜鸟的老菜鸟，为了祖国的编程事业，不惜拿出一个月工资，淘了一台配置稍高的二手笔记本，打算与老笔记本中的撸啊撸片彻底说再见，誓要在新机种开启一番撸啊撸的新事业。当然，撸代码离不开基本的环境搭建，每次新机上手都要百度一堆杂乱的似是而非的环境搭建教程，才勉强把开发环境搭建起来，时间耗费不少，但终是拾人牙慧，生搬硬套，从未能理解深层原理，将其内化吸收形成自己的技术积累；而环境搭建虽然基础，但编程界多少前辈大牛都是从此开始的，千万行代码的撸出也都依赖于此，所以，花时间总结一下，思考一番，确有必要。今天先来讲讲最最基本的**环境变量**，博主度娘无数，收获不小，现分两个方面来探究**环境变量**的问题：一是为什么要配置，二是如何配置。

配置环境变量:Why?

为了回答为什么的问题，还是先将JDK下载好，然后布衣博主先一步步的试错完毕，然后再跟大家分享正确的姿势。安装JDK首先肯定是去[Oracle官网](#)下载适合自己的开发版本啦，或许是沉浸在JDK1.7的开发环境中太久了，不看官网不知道，一看尼玛JDK1.9都发布了，可怜布衣博主连JDK1.8长啥样都还没见过呢。伤心三秒，果断下载JDK1.8的版本来尝尝。下面是下载主页的重点内容，相信大家都知道具体操作：

Which Java package do I need?

- **Software Developers: JDK** (Java SE Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- **Administrators running applications on a server: Server JRE** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#) ▶
- **End user running Java on a desktop: JRE:** (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

针对开发 (jdk) 以及
运行 (jre) 文字说
明, 对应着下载就可
以了

Java SE 8u161/ 8u162

Java SE 8u161 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u162 is a patch-set update including all of 8u161 plus additional bug fixes (described in the release notes).

[Learn more](#) ▶

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
 - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)
 - [JDK ReadMe](#)
 - [JRE ReadMe](#)

JDK

[DOWNLOAD](#) ▶

Server JRE

[DOWNLOAD](#) ▶

JRE

[DOWNLOAD](#) ▶

Java SE
Java EE
Java ME
Java SE Advanced & Suite
Java Embedded
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

[Overview](#)
[Downloads](#)
[Documentation](#)
[Community](#)
[Technologies](#)
[Training](#)

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

JDK 8u161 [checksum](#)

JDK 8u162 [checksum](#)

Java SE Development Kit 8u161

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

下载前必须同意

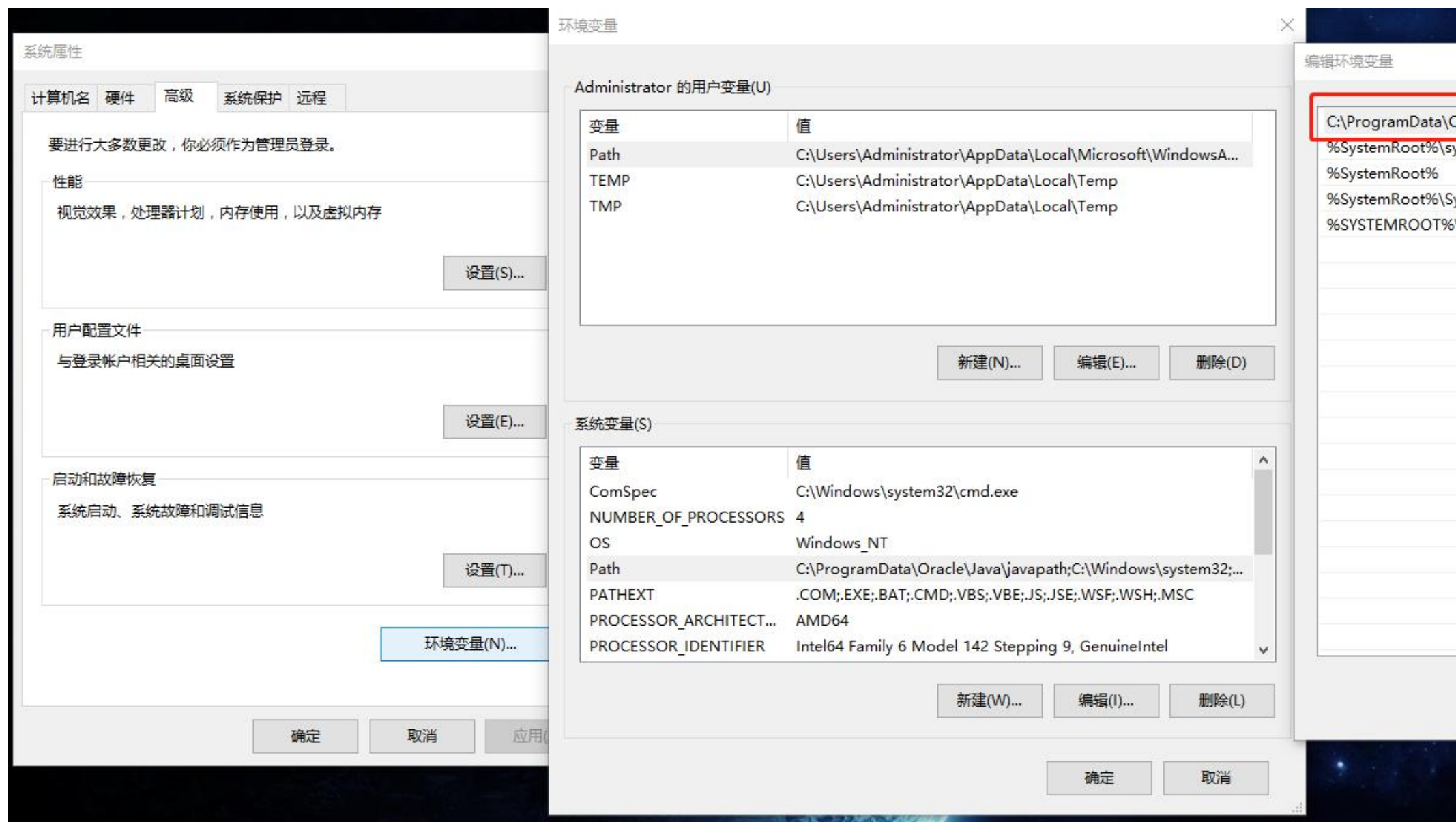
☒ Accept License Agreement

☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	jdk-8u161-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.88 MB	jdk-8u161-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.96 MB	jdk-8u161-linux-i586.rpm
Linux x86	183.76 MB	jdk-8u161-linux-i586.tar.gz
Linux x64	166.09 MB	jdk-8u161-linux-x64.rpm
Linux x64	180.97 MB	jdk-8u161-linux-x64.tar.gz
macOS	247.12 MB	jdk-8u161-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.99 MB	jdk-8u161-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.29 MB	jdk-8u161-solaris-sparcv9.tar.gz
Solaris x64	140.57 MB	jdk-8u161-solaris-x64.tar.Z
Solaris x64	97.02 MB	jdk-8u161-solaris-x64.tar.gz
Windows x86	198.54 MB	jdk-8u161-windows-i586.exe
Windows x64	206.51 MB	jdk-8u161-windows-x64.exe

针对不同操作系统的JDK下载包

安装JDK也很简单，先是JDK，然后是JRE,如果要自定义路径，注意两者不要安装在同一个包下面。然后，作为Java开发者，就会按基本套路开始配置起环境变量。如果仅仅是作为一个代码搬运工，百度有一堆环境变量的配置方式，傻瓜也会配置，但要想成为一个高阶一些的码农，在进行一个操作之前，总该知道你干这件事的缘由——为什么要配置环境变量，如果不配置又会怎样？平时看起来只是开发入门的基础操作，深究起来其实并不简单。有人会呵呵一笑很倾城——不配环境变量，我们怎么开发？欸，这样的问题，我只能说，编程这玩意儿，很多时候不动脑子是可以的，但是不动手却不行。布衣博主以前的固有思想也是觉得，配置环境变量是Java开发的固定姿势，可是今天为了深究环境变量这个问题，下载JDK1.8来折腾一番后发现，仅仅是安装好JDK和JRE，**根本用不着配置什么环境变量就可以开发和跑项目了**，这是怎么回事？到环境变量配置页查看，会发现在path的值中有个C:\ProgramData\Oracle\Java\javapath路径值生成，如以下图中展示（注：图例一为Win10系统的环境变量配置界面，和Win7有很大不同）——



我的电脑 > 系统 (C:) > ProgramData > Oracle > Java > javapath

名称	修改日期	类型	大小
java.exe	2018/3/18 12:10	应用程序	203 KB
javaw.exe	2018/3/18 12:10	应用程序	203 KB
javaws.exe	2018/3/18 12:10	应用程序	336 KB

JDK1.8的JRE安装成功后在自动生成的环境变量映射路径下已经有了可执行文件
所以我们在cmd中能够执行java命令

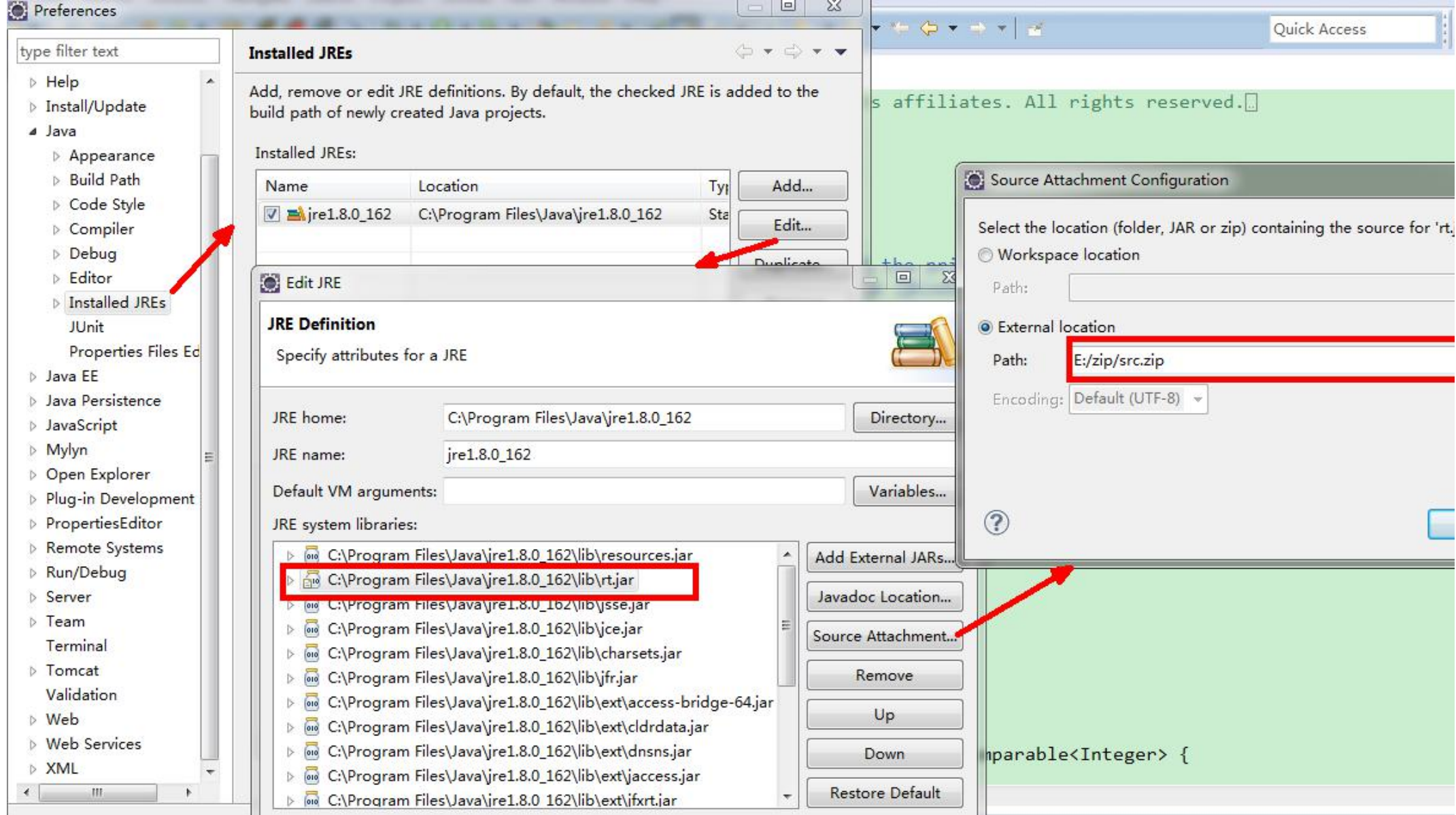
管理员: C:\Windows\system32\cmd.exe

Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>java -version
java version "1.8.0_162"
Java(TM) SE Runtime Environment (build 1.8.0_162-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)

C:\Users\Administrator>

而在映射的路径文件夹中已经存在了可执行的java.exe,这似乎是说明新版JDK（1.8）在安装的时候就已经自动为我们添加好了环境配置。当博主删除该path映射的路径值C:\ProgramData\Oracle\Java\javapath的时候，集成开发环境中的Eclipse/MyEclipse就无法正常打开了，提示需要JDK或JRE,但是IDEA依然能正常使用，恢复之后Eclipse/MyEclipse能够正常使用。接着将JDK1.8卸载，仅保留运行环境JRE，这个时候IDEA少了JDK的支持就无法正常开发使用了，而Eclipse/MyEclipse却依然可以正常使用，这是因为Eclipse/MyEclipse比较例外的是它自己内嵌了Java编译器，对javac没有任何依赖关系,所以Eclipse/Myeclipse其实只需要依赖JRE就可以做开发并编译运行Java文件的，有了JDK只是为了方便查询源码——其实我们不用安装JDK同样可以查看源码，你只要我们的源码包放在任意路径，然后在设置中绑定源码包就可以了，如下图——



接着卸载JRE,这个时候开发工具肯定是没法使用了,关键是在环境变量的Path值C:\ProgramData\Oracle\Java\javapath也消失了,对应路径下的文件也被删除了,这就说明在安装JRE的时候,是会为我们自动配置好环境变量的。环境变量都没配置就成功的布衣博主实在不甘心,卸载了JDK1.8之后又单独安装了JDK1.7继续尝试,安装完成后Path值并没有自动添加,博主也没有主动的去配置任何环境变量,但是打开Eclipse或者Idea等IDE,依然可以正常开发跑项目。尝试至此,似乎证明了长久以来大家配置环境变量只是思维和行为固化后的自以为是,纯粹是多此一举,对日常开发调试并没有卵用。难道,这么多程序员都在信奉了一个错误的认知而且丝毫不觉得有异?

要搞解开这个疑惑,还是得搞清楚环境变量到底是干什么的,为什么要配置它。所谓环境变量,其实就是操作系统中用来指定操作系统运行环境的一些参数,比如临时文件夹位置和系统文件夹位置等。当要求系统运行一个程序而没有告诉它程序所在的完整路径时,系统除了在当前目录下面寻找此程序外,还应到环境变量的PATH中指定的路径值去找。也就是说当执行可执行文件找不到位置的时候,就会去PATH中配置的路径去找。这里你可以做个小测试,就是任意

位置新建一个哪怕普通文本文件，将其路径配置成环境变量，然后通过cmd命令行输入文件名，都能打开那个文件，这就环境变量的功劳。因此可以设想如果直接在可执行文件所在文件夹取执行，那不就是找得到吗？为了证明，博主先打开cmd执行编译命令javac，返回不是内部或外部命令，当我们切换到JDK的bin目录C:\Program Files\Java\jdk1.8.0_162\bin路径去执行javac命令的时候，就能够成功执行，好像配置了环境变量一样——



既然可以执行javac，那么我们就在该目录下测试编译一个Java文件。先用记事本编写测试代码 `public class Test { public static void main(String[] args) { System.out.println("博客园 陈本布衣"); } }`，文件名必须和类名保持一致Test.java,然后我们执行编译命令并运行文件——

C:\Program Files\Java\jdk1.8.0_162\bin

共享 查看

我的电脑 > 系统 (C:) > Program Files > Java > jdk1.8.0_162 > bin

名称	修改日期	类型	大小
Test.class	2018/3/18 13:41	CLASS 文件	1 KB
Test.java	2018/3/18 13:37	IntelliJ IDEA	1 KB
appletviewer.exe	2018/3/18 12:09	应用程序	16 KB
extcheck.exe	2018/3/18 12:09	应用程序	17 KB
idlj.exe	2018/3/18 12:09	应用程序	17 KB
jabswitch.exe	2018/3/18 12:09	应用程序	34 KB
jar.exe	2018/3/18 12:09	应用程序	16 KB
jarsigner.exe	2018/3/18 12:09	应用程序	17 KB
java.exe	2018/3/18 12:09	应用程序	203 KB
javac.exe	2018/3/18 12:09	应用程序	16 KB
javadoc.exe	2018/3/18 12:09	应用程序	17 KB
javafxpackager.exe	2018/3/18 12:09	应用程序	146 KB
javah.exe	2018/3/18 12:09	应用程序	16 KB
javap.exe	2018/3/18 12:09	应用程序	16 KB
javapackager.exe	2018/3/18 12:09	应用程序	146 KB
java-rmi.exe	2018/3/18 12:09	应用程序	16 KB
javaws.exe	2018/3/18 12:09	应用程序	203 KB

1 个项目 125 字节

执行编译命令后产生

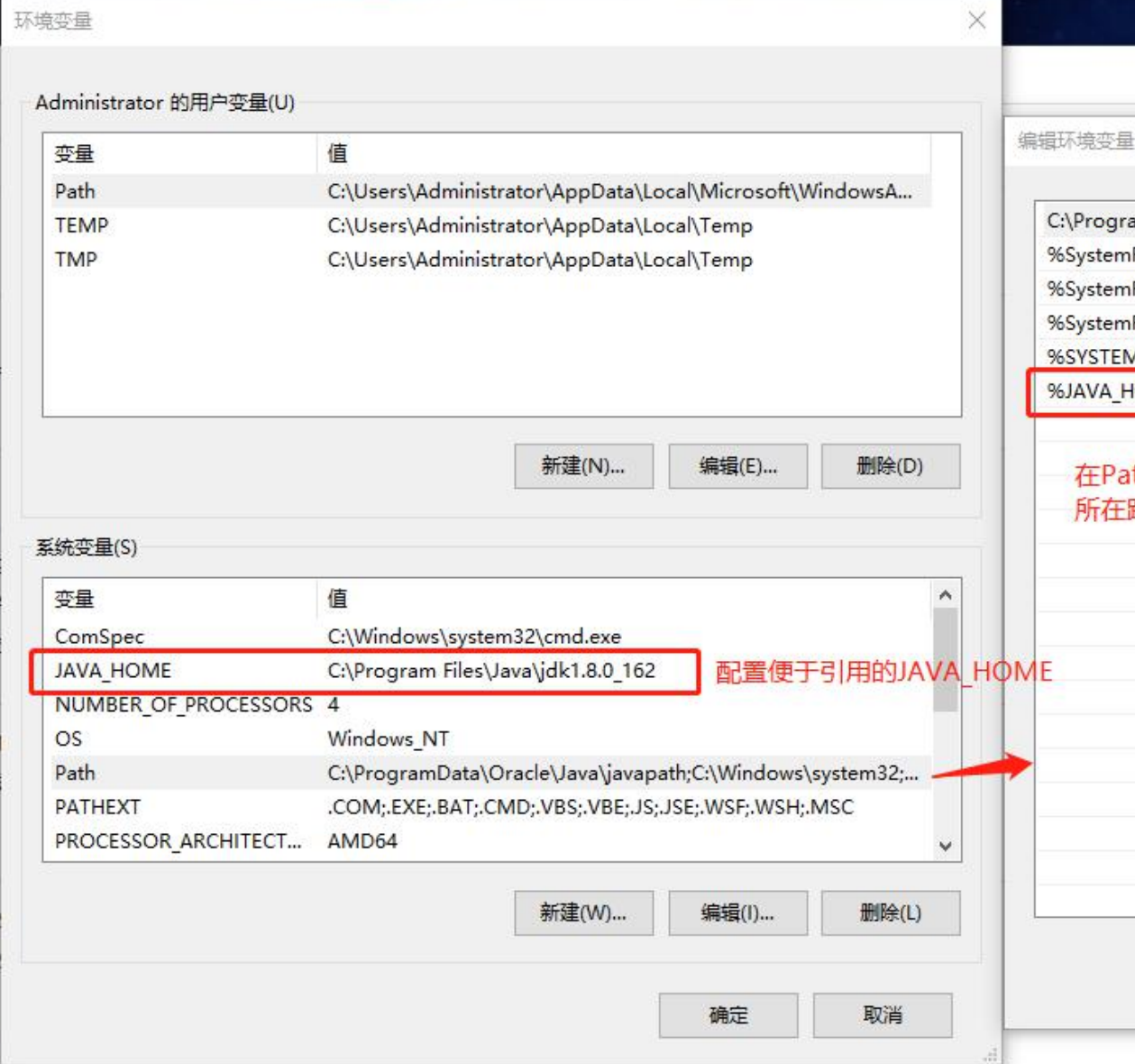
管理员: C:\Windows\System32\cmd.exe

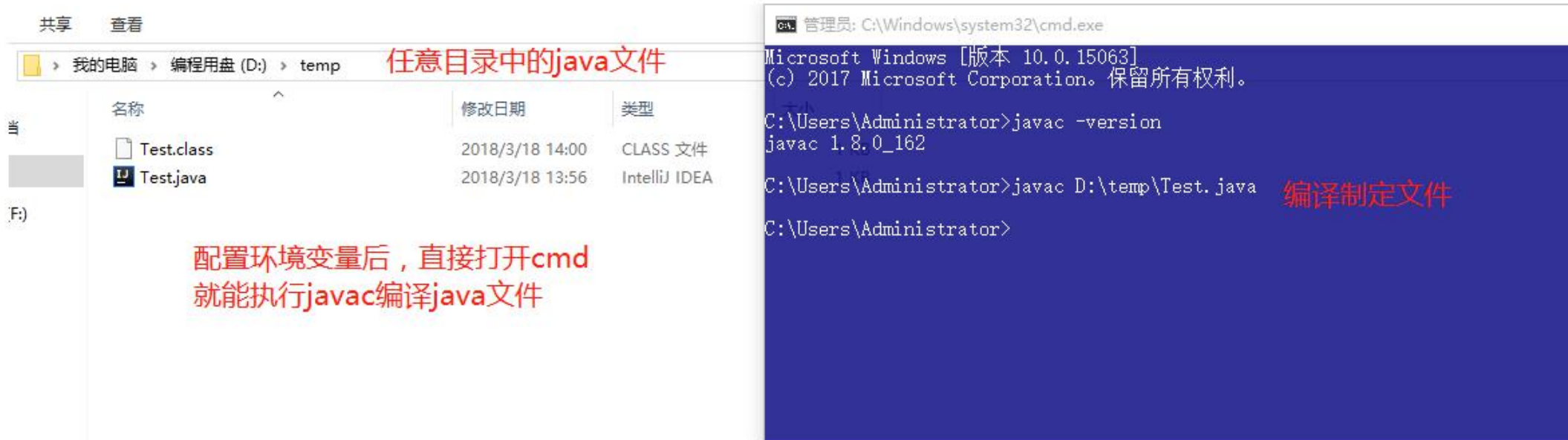
```
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Program Files\Java\jdk1.8.0_162\bin>javac T
C:\Program Files\Java\jdk1.8.0_162\bin>java Te
博客园 陈本布衣

C:\Program Files\Java\jdk1.8.0_162\bin>
```

可是正常情况下我的Java文件是不可能放在bin目录下的，所以我们接着配置好环境变量，使我们能够在任意目录编译Java文件——





配置环境变量的目的似乎达到了，仅仅是为了不用切换到JDK的bin目录就能执行编译命令的方便，可是，现在都是用集成开发环境如idea或eclipse等开发并编译Java程序，如果不是使用命令行javac的方式来编译Java程序，那么，广大的程序员们，你在搭建开发环境的时候，配置环境变量到底是因为什么？

配置环境变量:How?

欸，本来前文阐述了环境变量的原理后，这里计划该写如何配置了，但是布衣博主在不断尝试之后发现环境变量在Java的开发环境中根本不是必须的配置，所以，这里的How就只能戛然而止了，实在不想多叨叨如何配置那不必要的玩意儿了。当然，或许我前文的尝试和总结很有问题，因为配置了多年环境变量的博主自己一直都还在怀疑上文的尝试是否正确，也**诚心希望各路大神指出阐述中的问题**，共同探讨环境变量配置的必要性和可用之处，以免误导他人；

原创博文，转载请注明出处和原文链接，谢谢！

作者：陈本布衣