

怎样配置 Java 开发环境？详解（附 JDK 快速下载方式）

配置 Java 开发环境

先了解三个概念

JVM

JVM(Java Virtual Machine, Java 虚拟机)是一种具备计算机核心配置的计算环境,在实际的计算机上虚拟计算机各种运算功能,有自己完善的架构,如处理器、堆栈、寄存器等,还具有相应的指令系统。且 JVM 屏蔽了与具体操作系统平台相关的信息,使得 Java 程序只需生成在 Java 虚拟机上运行的目标代码(字节码),就可以在多种平台上不加修改地运行。

JRE

JRE(Java Runtime Environment, Java 运行环境)是运行 JAVA 程序所必须的环境,包含 JVM 标准实现及 Java 核心类库,如果只需要运行 Java 程序,安装 JRE 即可。

JDK

JDK (Java Development Kit , Java 开发工具包)

JDK=JRE+Java 的开发工具[java, javac, javadoc, javap 等], JDK 是提供给程序员开发 Java 程序的开发工具包,开发 Java 程序前必须安装 JDK,包含了 Java 的开发工具,也包括了 JRE,所以安装了 JDK,就不要再安装 JRE。

什么是 java 开发环境？

任何一种语言都需要运行环境和开发环境,对于 Java 开发者来讲,Java 开发环境就是 JDK,运行环境就是 JRE,需要在系统的环境变量配置 JDK、JRE 路径。JDK 环境变量配置的作用主要是让 Java 虚拟机能进行自动编译 class 文件,因为 Java 很多可执行文件都是在 JDK 的 bin 目录下,当执行要编译.java 文件时,自动寻找 javac.exe,而 javac.exe 又去寻找其他的可执行文件,直到.java 文件编译成.class 文件。编译结束后通过 java 命令运行.class 文件。

Java 开发环境配置步骤

以常用的 window10 64 系统,Java1.8 版本为例

1 下载并安装 JDK

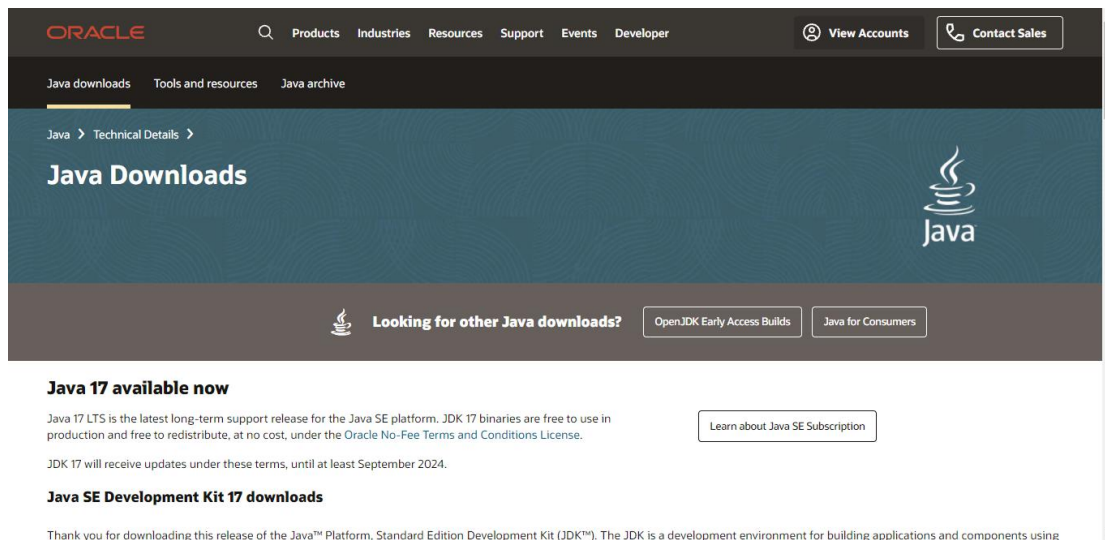
1.1 下载 JDK

访问 oracle 官网

•

<https://www.oracle.com/java/technologies/javase-downloads.html>

下载 JDK1.8 即可(需要注册 oracle 账号,下载速度稍慢,文末附快速下载方式)



1.2 安装 JDK



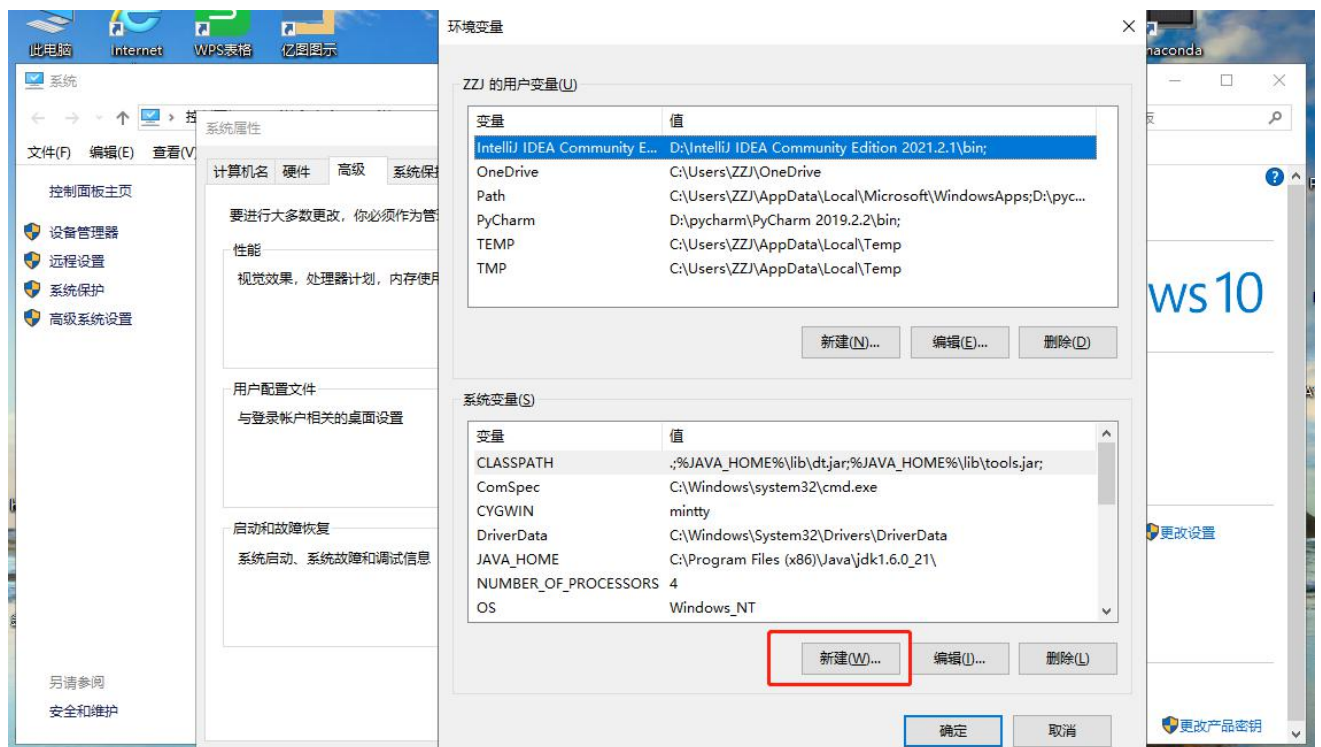
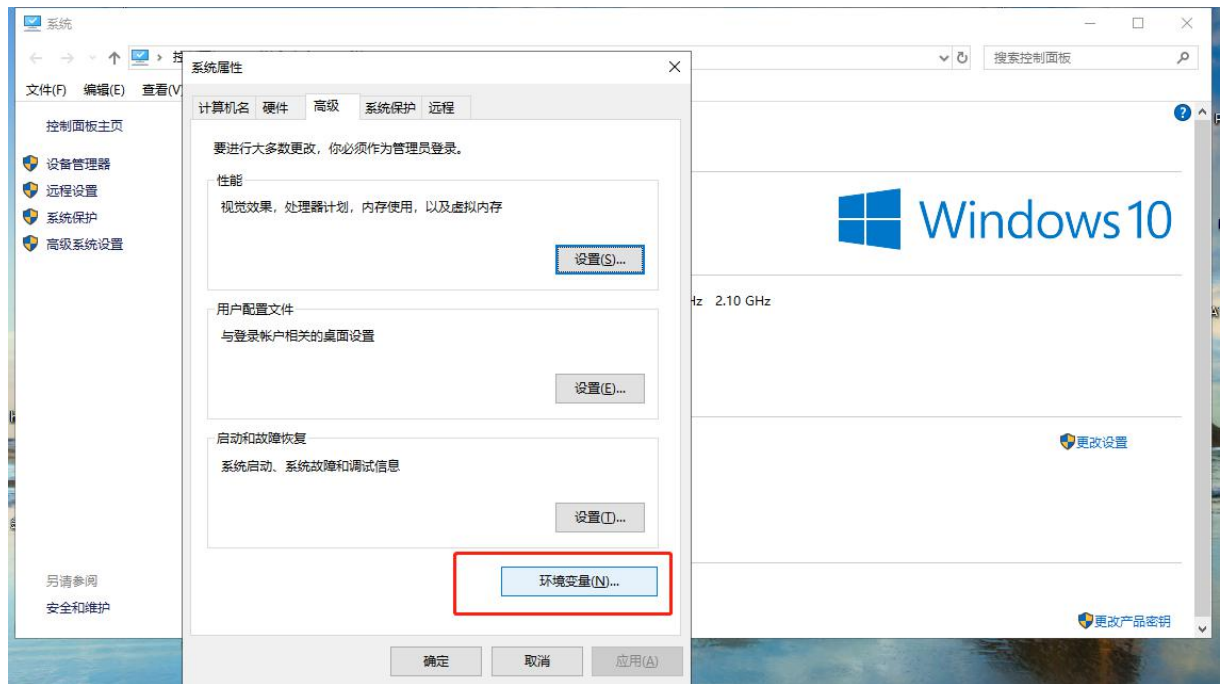
2 配置 JDK 环境

JDK 安装完成, 计算机已经存在 Java 开发环境, 但是计算机还未与 JDK 融为一体, 无法使用 Java 开发工具, 还需要进一步配置 Java 开发环境。

2.1 进入环境配置界面

右键“此电脑”单击“属性”，点击“高级系统设置”，点击“环境变量”即可“新建”如下：

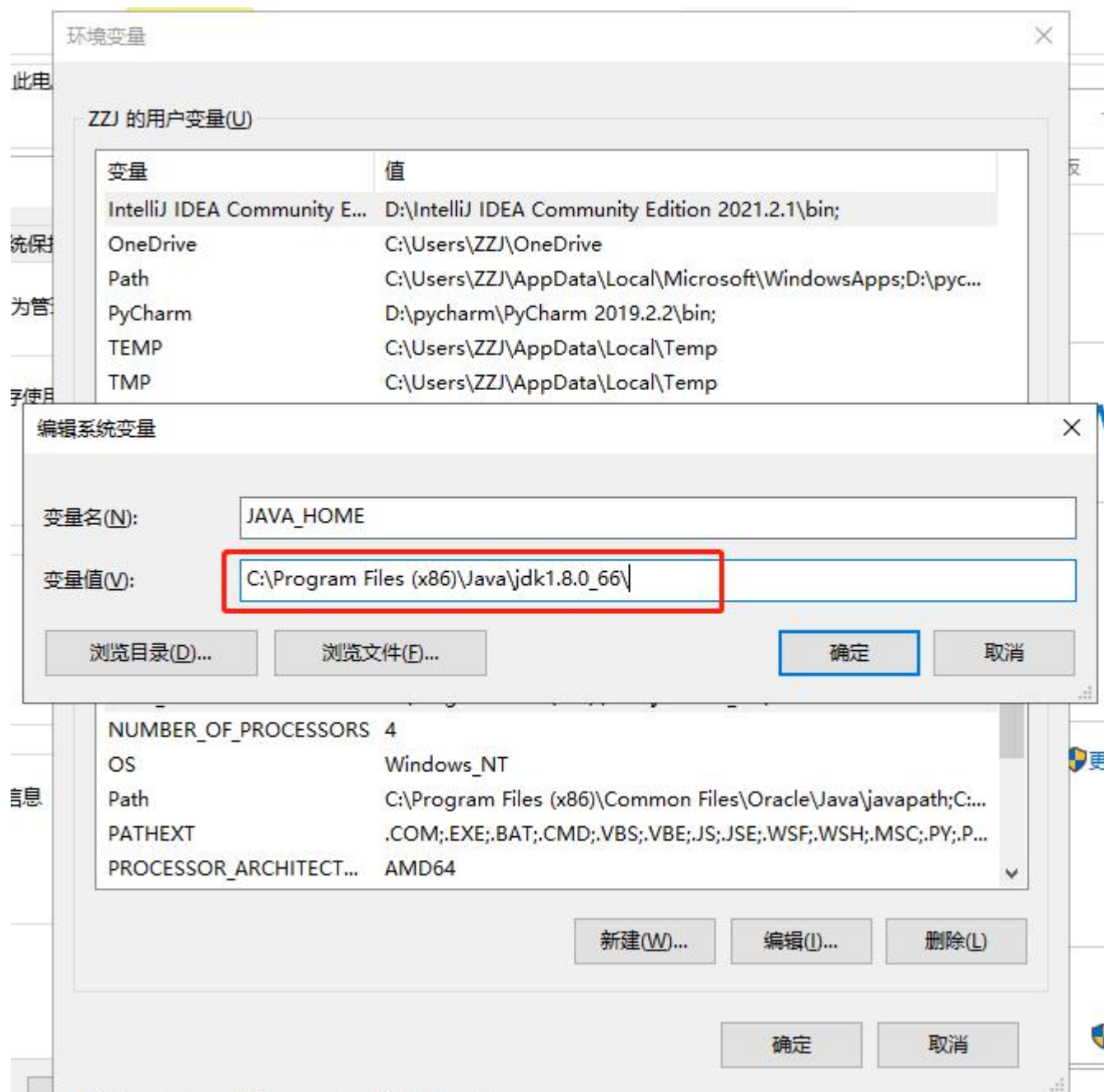




（用户变量和系统变量的差别：在用户变量下新建环境变量，即只能在该用户下使用此环境；在系统变量下新建环境变量，可以在所有用户下使用此环境，一般默认在系统变量下新建环境变量）

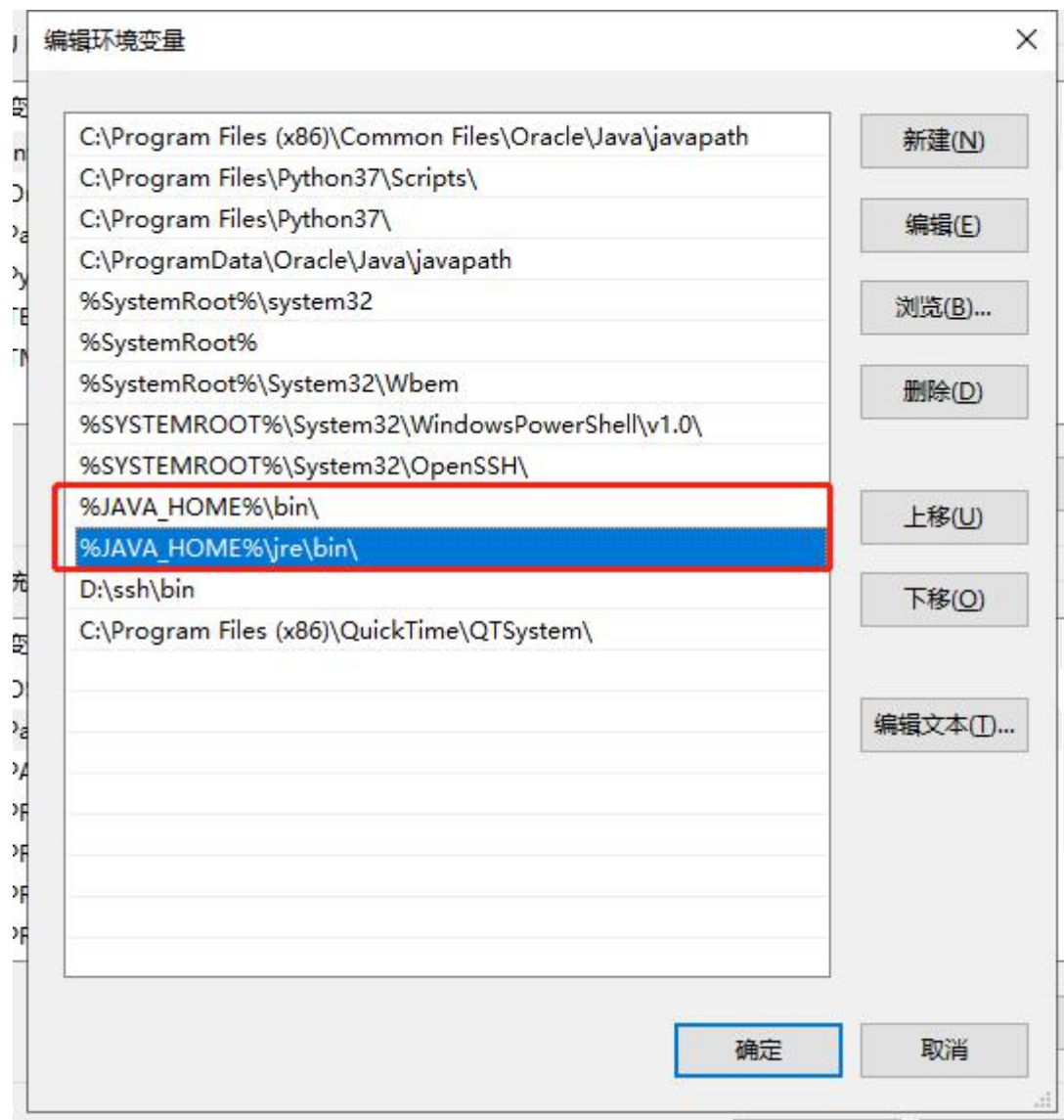
2.2 JAVA_HOME

新建环境变量变量名：“**JAVA_HOME**”，变量值：“**C:\Program Files (x86)\Java\jdk1.8.0_66**”（安装 JDK 的路径）



2.3Path

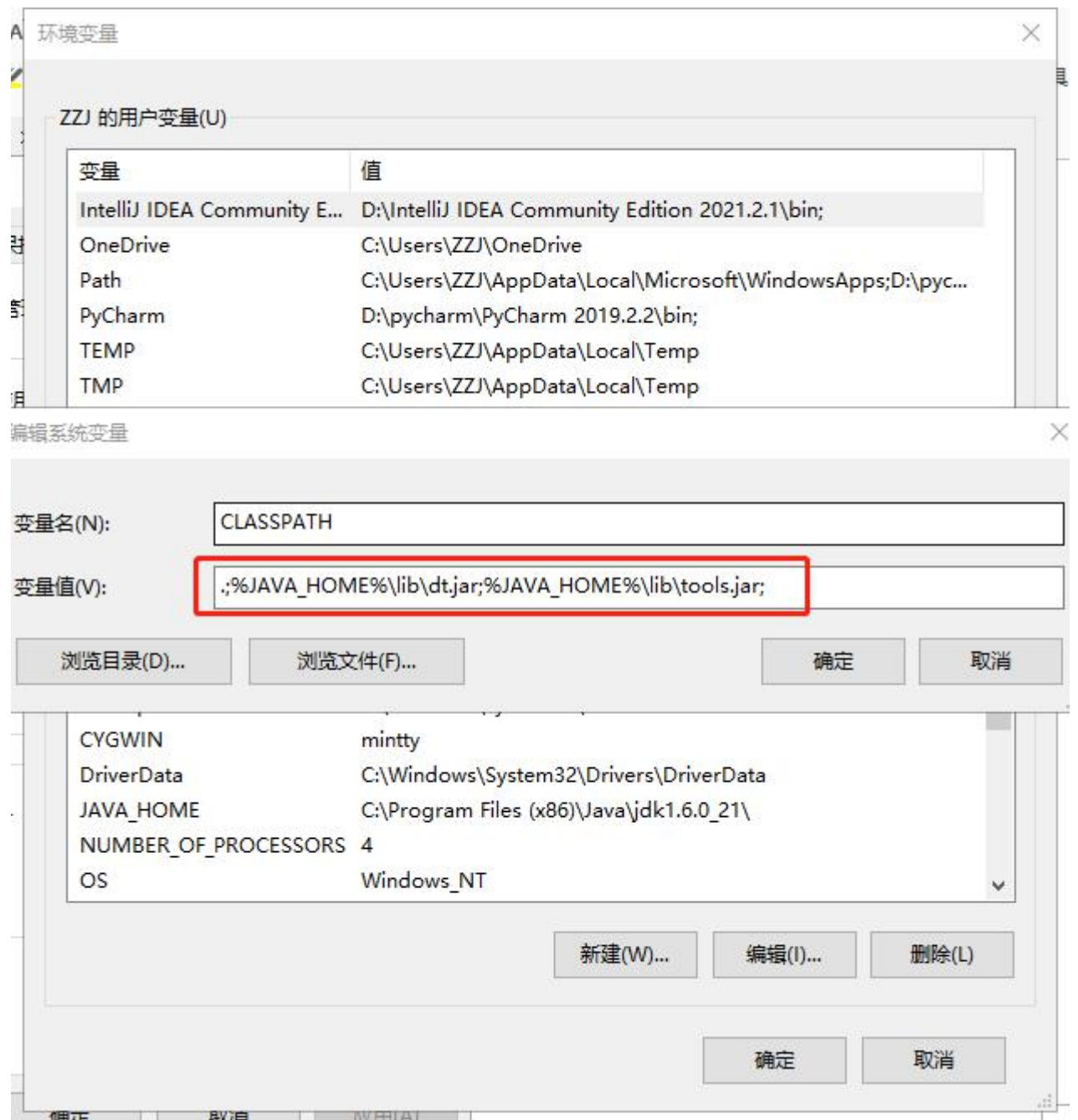
编辑 Path 环境变量，新建一个变量值：“**%JAVA_HOME%\bin;** ”；再
新建一个变量值：“**%JAVA_HOME%\bin;** ”



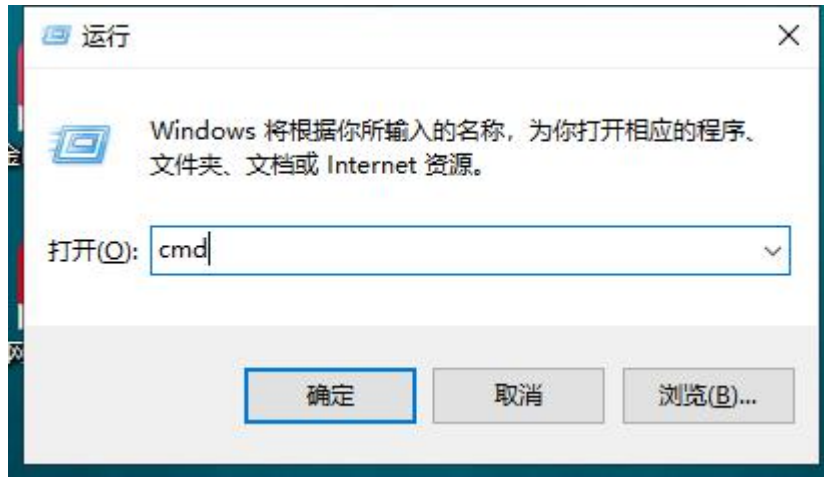
2.4 CLASSPATH

新建环境变量变量名：“CLASSPATH”，变量值：

“.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;”，确定即可



3 验证 Java 开发环境



Windows 键+R, 输入“**cmd**”, 确定进入命令提示符界面, 开始如下验证。

3.1Java 版本

输入“**java - version**”

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.17763.1879]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\ZZJ>java -version
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 1.8.0_301-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.301-b09, mixed mode)

C:\Users\ZZJ>
```

3.2 编译命令

输入“**javac**”

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.17763.1879]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\ZZJ>java -version
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 1.8.0_301-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.301-b09, mixed mode)

C:\Users\ZZJ>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are used
  -classpath <path>  Specify where to find user class files and annotation processors
  -cp <path>        Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>    Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:{none,only} Control whether annotation processing and/or compilation is done.
  -processor <class1>[, <class2>...<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>      Specify where to place generated class files
  -s <directory>      Specify where to place generated source files
  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
```

3.3 运行命令

输入“java”

```
C:\Windows\system32\cmd.exe
-d <directory>      Specify where to place generated class files
-s <directory>      Specify where to place generated source files
-implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
-encoding <encoding> Specify character encoding used by source files
-source <release>    Provide source compatibility with specified release
-target <release>    Generate class files for specific VM version
-version            Version information
-help              Print a synopsis of standard options
-Akey[=value]       Options to pass to annotation processors
-X                 Print a synopsis of nonstandard options
-J<flag>            Pass <flag> directly to the runtime system

C:\Users\ZZJ>java
用法: java [-options] class [args...]
           (执行类)
或 java [-options] -jar jarfile [args...]
           (执行 jar 文件)
其中选项包括:
  -d32             使用 32 位数据模型 (如果可用)
  -d64             使用 64 位数据模型 (如果可用)
  -server          选择 "server" VM
                  默认 VM 是 server.

  -cp <目录和 zip/jar 文件的类搜索路径>
  -classpath <目录和 zip/jar 文件的类搜索路径>
              用 ; 分隔的目录, JAR 档案
              和 ZIP 档案列表, 用于搜索类文件。

  -D<名称>=<值>    设置系统属性
```

验证成功，配置完成

JDK 软件快速下载及电子版 Java 环境配置方法获取方式:

关注微信公众号“Java 烂笔头”，回复“Java 环境”

