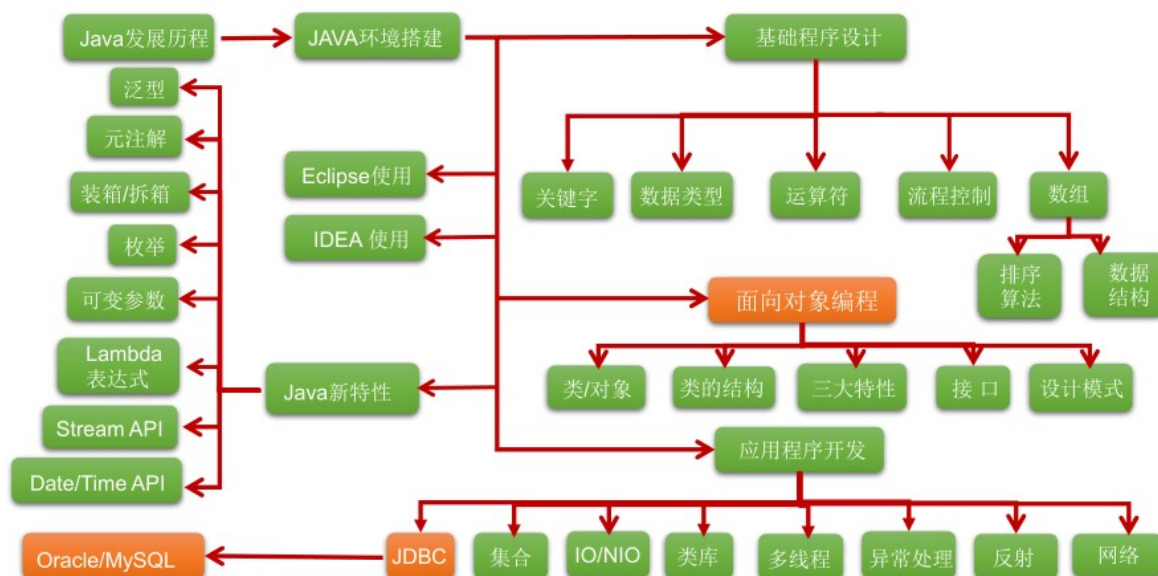


Java基础是学习JavaEE、大数据、Android开发的基石！

Java基础知识图解：



## 一、软件开发介绍

### ● 软件开发

软件，即一系列按照特定顺序组织的计算机数据和指令的集合。有系统软件和应用软件之分。

### ● 人机交互方式

- 图形化界面(Graphical User Interface GUI)这种方式简单直观，使用者易于接受，容易上手操作。
- 命令行方式(Command Line Interface CLI)：需要有一个控制台，输入特定的指令，让计算机完成一些操作。较为麻烦，需要记录住一些命令。

Pascal之父Nicklaus Wirth: "Algorithms+Data Structures=Programs"

## ●常用的DOS命令

- **dir** : 列出当前目录下的文件以及文件夹
  - **md** : 创建目录
  - **rd** : 删除目录
  - **cd** : 进入指定目录
  - **cd..** : 退回到上一级目录
  - **cd\** : 退回到根目录
  - **del** : 删除文件
  - **exit** : 退出 dos 命令行
- ✓ 补充: `echo javase>1.doc`

## ● 常用快捷键

- ← →: 移动光标
- ↑ ↓: 调阅历史操作命令
- Delete和Backspace: 删除字符

# 二、计算机编程语言介绍

---

## ●什么是计算机语言

- 语言: 是人与人之间用于沟通的一种方式。例如: 中国人与中国人用普通话沟通。而中国人要和英国人交流, 就要学习英语。

### ➤ 计算机语言: 人与计算机交流的方式。

如果人要与计算机交流, 那么就要学习计算机语言。

计算机语言有很多种。如: **C ,C++ ,Java ,PHP , Kotlin, Python, Scala**等。

## ● 第一代语言

- 机器语言。指令以二进制代码形式存在。

## ● 第二代语言

- 汇编语言。使用助记符表示一条机器指令。



## ● 第三代语言：高级语言

- C、Pascal、Fortran面向过程的语言
- C++面向过程/面向对象
- **Java**跨平台的纯面向对象的语言
- .NET跨语言的平台
- Python、Scala...

# 三、Java语言概述

---

## 1、概述

---

- 是SUN(**S**tanford **U**niversity **N**etwork, 斯坦福大学网络公司 ) 1995年推出的一门高级编程语言。
- 是一种面向Internet的编程语言。Java一开始富有吸引力是因为Java程序可以在Web浏览器中运行。这些Java程序被称为Java小程序（applet）。applet使用现代的图形用户界面与Web用户进行交互。 applet内嵌在HTML代码中。
- 随着Java技术在web方面的不断成熟，已经成为Web应用程序的首选开发语言。

后台开发：Java、PHP、Python、Go、Node.js

## 2、Java 简史

---

- 1991年 Green项目，开发语言最初命名为Oak (橡树)
- 1994年，开发组意识到Oak 非常适合于互联网
- 1996年，发布JDK 1.0，约8.3万个网页应用Java技术来制作
- 1997年，发布JDK 1.1， JavaOne会议召开，创当时全球同类会议规模之最
- 1998年，发布JDK 1.2，同年发布企业平台J2EE
- 1999年，Java分成J2SE、J2EE和J2ME，JSP/Servlet技术诞生
- 2004年，**发布里程碑式版本：JDK 1.5，为突出此版本的重要性，更名为JDK 5.0**
- 2005年，J2SE -> JavaSE，J2EE -> JavaEE，J2ME -> JavaME
- 2009年，Oracle公司收购SUN，交易价格74亿美元
- 2011年，发布JDK 7.0
- **2014年，发布JDK 8.0，是继JDK 5.0以来变化最大的版本**
- 2017年，发布JDK 9.0，最大限度实现模块化
- 2018年3月，发布JDK 10.0，版本号也称为18.3
- 2018年9月，发布JDK 11.0，版本号也称为18.9

让天下没有难学的技术

## 3、Java 技术体系平台

---

<b>Java SE(Java Standard Edition)标准版</b>
支持面向桌面级应用（如Windows下的应用程序）的Java平台，提供了完整的Java核心API，此版本以前称为J2SE
<b>Java EE(Java Enterprise Edition)企业版</b>
是为开发企业环境下的应用程序提供的一套解决方案。该技术体系中包含的技术如：Servlet、Jsp等，主要针对于Web应用程序开发。版本以前称为J2EE
<b>Java ME(Java Micro Edition)小型版</b>
支持Java程序运行在移动终端（手机、PDA）上的平台，对Java API有所精简，并加入了针对移动终端的支持，此版本以前称为J2ME
<b>Java Card</b>
支持一些Java小程序（Applets）运行在小内存设备（如智能卡）上的平台

## 4、Java 在各领域的应用

- 从Java的应用领域来分，Java语言的应用方向主要表现在以下几个方面：
  - **企业级应用**：主要指复杂的大企业的软件系统、各种类型的网站。Java的安全机制以及它的跨平台的优势，使它在分布式系统领域开发中有广泛应用。应用领域包括金融、电信、交通、电子商务等。
  - **Android平台应用**：Android应用程序使用Java语言编写。Android开发水平的高低很大程度上取决于Java语言核心能力是否扎实。
  - **大数据平台开发**：各类框架有Hadoop, spark, storm, flink等，就这类技术生态圈来讲，还有各种中间件如flume, kafka, sqoop等等，这些框架以及工具大多数是用Java编写而成，但提供诸如Java, scala, Python, R等各种语言API供编程。
  - **移动领域应用**：主要表现在消费和嵌入式领域，是指在各种小型设备上的应用，包括手机、PDA、机顶盒、汽车通信设备等。

## 5、Java 语言的诞生

java之父James Gosling团队在开发“Green”项目时，发现C缺少垃圾回收系统，还有可移植的安全性、分布程序设计和多线程功能。最后，他们想要一种易于移植到各种设备上的平台。

Java确实是从C语言和C++语言继承了许多成份，甚至可以将Java看成是**类C语言**发展和衍生的产物。比如Java语言的变量声明，操作符形式，参数传递，流程控制等方面和C语言、C++语言完全相同。但同时，Java是一个**纯粹的面向对象**的程序设计语言，它继承了C++语言面向对象技术的核心。Java**舍弃了C语言中容易引起错误的指针**（以引用取代）、运算符重载（operator overloading）、多重继承（以接口取代）等特性，**增加了垃圾回收器功能**用于回收不再被引用的对象所占据的内存空间。JDK1.5又引入了泛型编程（Generic Programming）、类型安全的枚举、不定长参数和自动装/拆箱



## 6、Java 语言的主要特性



- **Java语言是易学的。**Java语言的语法与C语言和C++语言很接近，使得大多数程序员很容易学习和使用Java。
- **Java语言是强制面向对象的。**Java语言提供类、接口和继承等原语，为了简单起见，只支持类之间的单继承，但支持接口之间的多继承，并支持类与接口之间的实现机制（关键字为implements）。
- **Java语言是分布式的。**Java语言支持Internet应用的开发，在基本的Java应用编程接口中有一个网络应用编程接口（java net），它提供了用于网络应用编程的类库，包括URL、URLConnection、Socket、ServerSocket等。Java的RMI（远程方法激活）机制也是开发分布式应用的重要手段。
- **Java语言是健壮的。**Java的强类型机制、异常处理、垃圾的自动收集等是Java程序健壮性的重要保证。对指针的丢弃是Java的明智选择。
- **Java语言是安全的。**Java通常被用在网络环境中，为此，Java提供了一个安全机制以防恶意代码的攻击。如：安全防范机制（类ClassLoader），如分配不同的名字空间以防替代本地的同名类、字节代码检查。
- **Java语言是体系结构中立的。**Java程序（后缀为java的文件）在Java平台上被编译为体系结构中立的字节码格式（后缀为class的文件），然后可以在实现这个Java平台的任何系统中运行。
- **Java语言是解释型的。**如前所述，Java程序在Java平台上被编译为字节码格式，然后可以在实现这个Java平台的任何系统的解释器中运行。
- **Java是性能略高的。**与那些解释型的高级脚本语言相比，Java的性能还是较优的。
- **Java语言是原生支持多线程的。**在Java语言中，线程是一种特殊的对象，它必须由Thread类或其子（孙）类来创建。

## 四、Java程序运行机制及运行过程

---

### 1、Java语言的特点

---

#### ➤特点一：面向对象

- ✓两个基本概念：类、对象
- ✓三大特性：封装、继承、多态

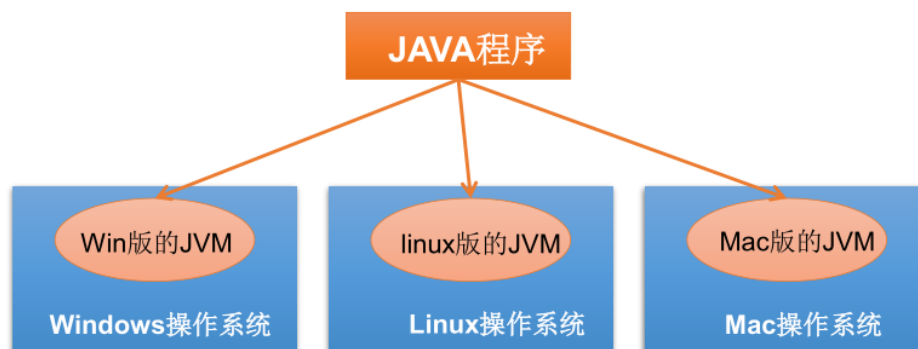
#### ➤特点二：健壮性

- ✓吸收了C/C++语言的优点，但去掉了其影响程序健壮性的部分（如指针、内存的申请与释放等），提供了一个相对安全的内存管理和访问机制

#### ➤特点三：跨平台性

- ✓跨平台性：通过Java语言编写的应用程序在不同的系统平台上都可以运行。“**Write once , Run Anywhere**”
- ✓原理：只要在需要运行 java 应用程序的操作系统上，先安装一个Java虚拟机 (JVM **J**ava **V**irtual **M**achine) 即可。由JVM来负责Java程序在该系统中的运行。

## ● Java语言的特点：跨平台性



因为有了JVM，同一个Java 程序在三个不同的操作系统中都可以执行。这样就实现了Java 程序的跨平台性。

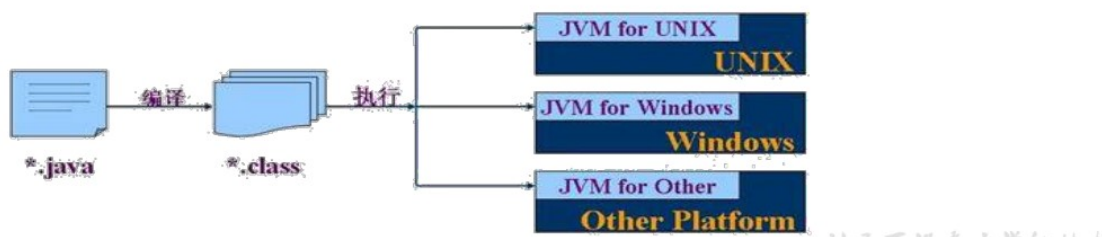
## 2、Java两种核心机制

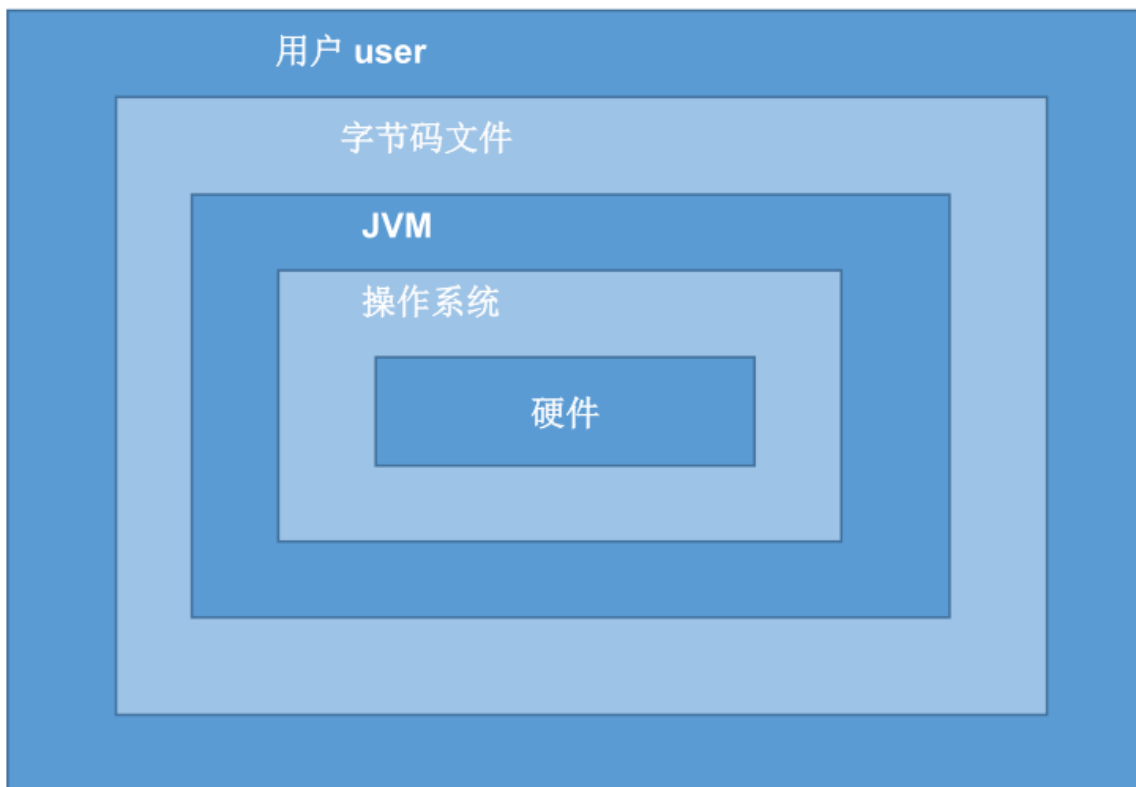
➤Java虚拟机 (Java Virtual Machine)

➤垃圾收集机制 (Garbage Collection)

### (1) 核心机制—Java 虚拟机

- JVM是一个虚拟的计算机，具有指令集并使用不同的存储区域。负责执行指令，管理数据、内存、寄存器。
- 对于不同的平台，有不同的虚拟机。
- 只有某平台提供了对应的java虚拟机，java程序才可在此平台运行
- Java虚拟机机制屏蔽了底层运行平台的差别，实现了“一次编译，到处运行”





## (2) 核心机制—垃圾回收

- 不再使用的内存空间应回收——垃圾回收。
  - 在C/C++等语言中，由程序员负责回收无用内存。
  - Java 语言消除了程序员回收无用内存空间的责任：它提供一种系统级线程跟踪存储空间的分配情况。并在JVM空闲时，检查并释放那些可被释放的存储空间。
- 垃圾回收在Java程序运行过程中自动进行，程序员无法精确控制和干预。

●Java程序还会出现内存泄漏和内存溢出问题吗？Yes!

# 五、Java的环境搭建

## 1、什么是JDK，JRE

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

JDK是提供给Java开发人员使用的，其中包含了java的开发工具，也包括了JRE。所以安装了JDK，就不用在单独安装JRE了。

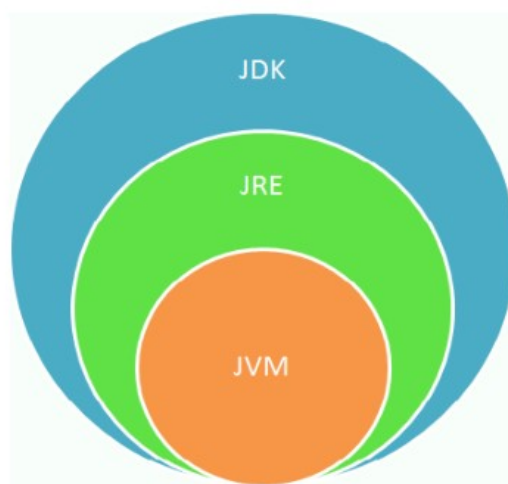
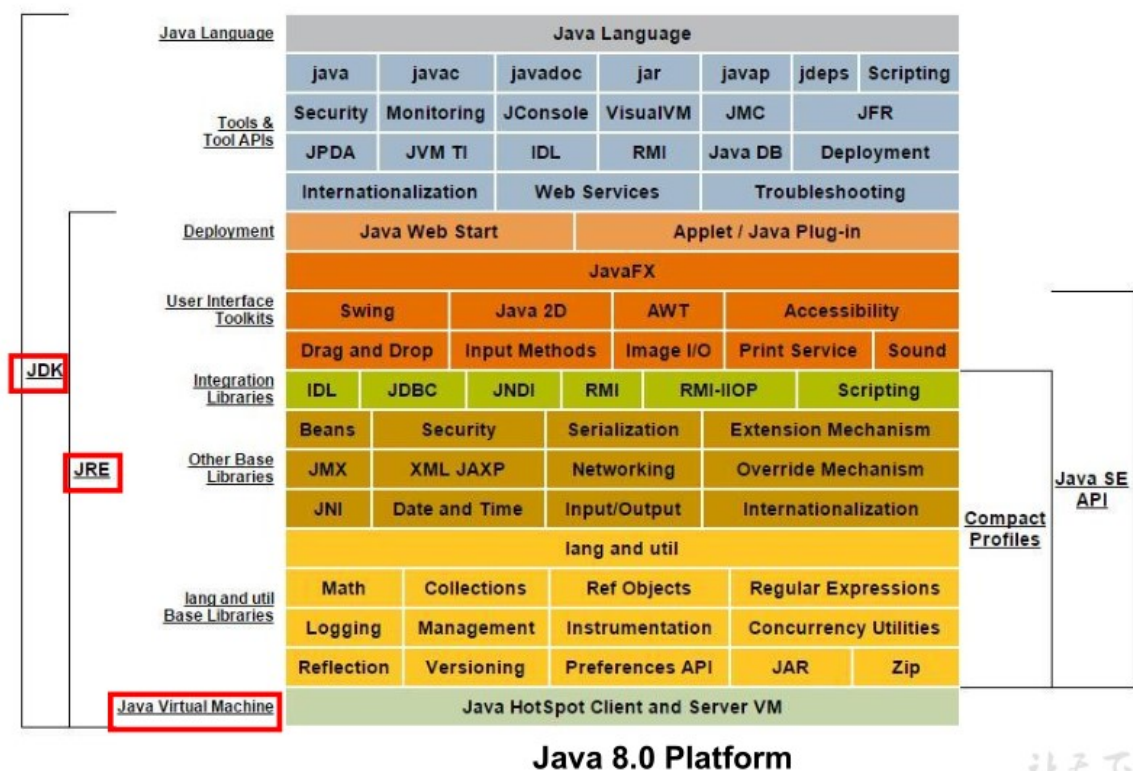
➢ 其中的开发工具：编译工具(javac.exe) 打包工具(jar.exe)等

### JRE(Java Runtime Environment Java运行环境)

包括Java虚拟机(JVM Java Virtual Machine)和Java程序所需的核心类库等，如果想要运行一个开发好的Java程序，计算机中只需要安装JRE即可。

简单而言，使用JDK的开发工具完成的java程序，交给JRE去运行。

## 2、JDK、JRE、JVM 关系



- JDK = JRE + 开发工具集（例如Javac编译工具等）
- JRE = JVM + Java SE标准类库

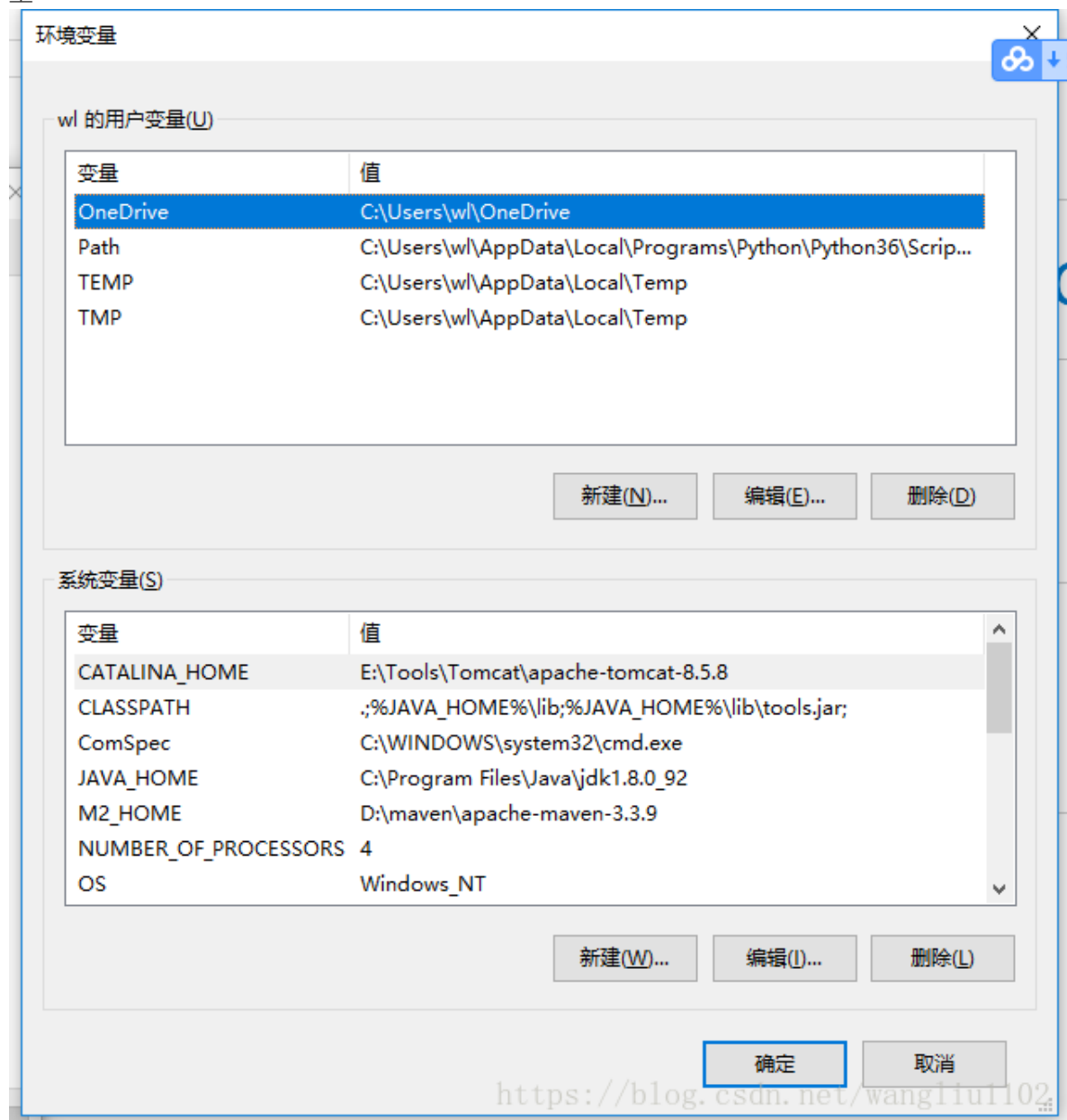
## 3、Windows中安装JDK

下载jdk-8u92-windows-x64.exe：链接：<https://pan.baidu.com/s/1P03MFxdsbNspjsRO10ufag> 提取码：p1m1

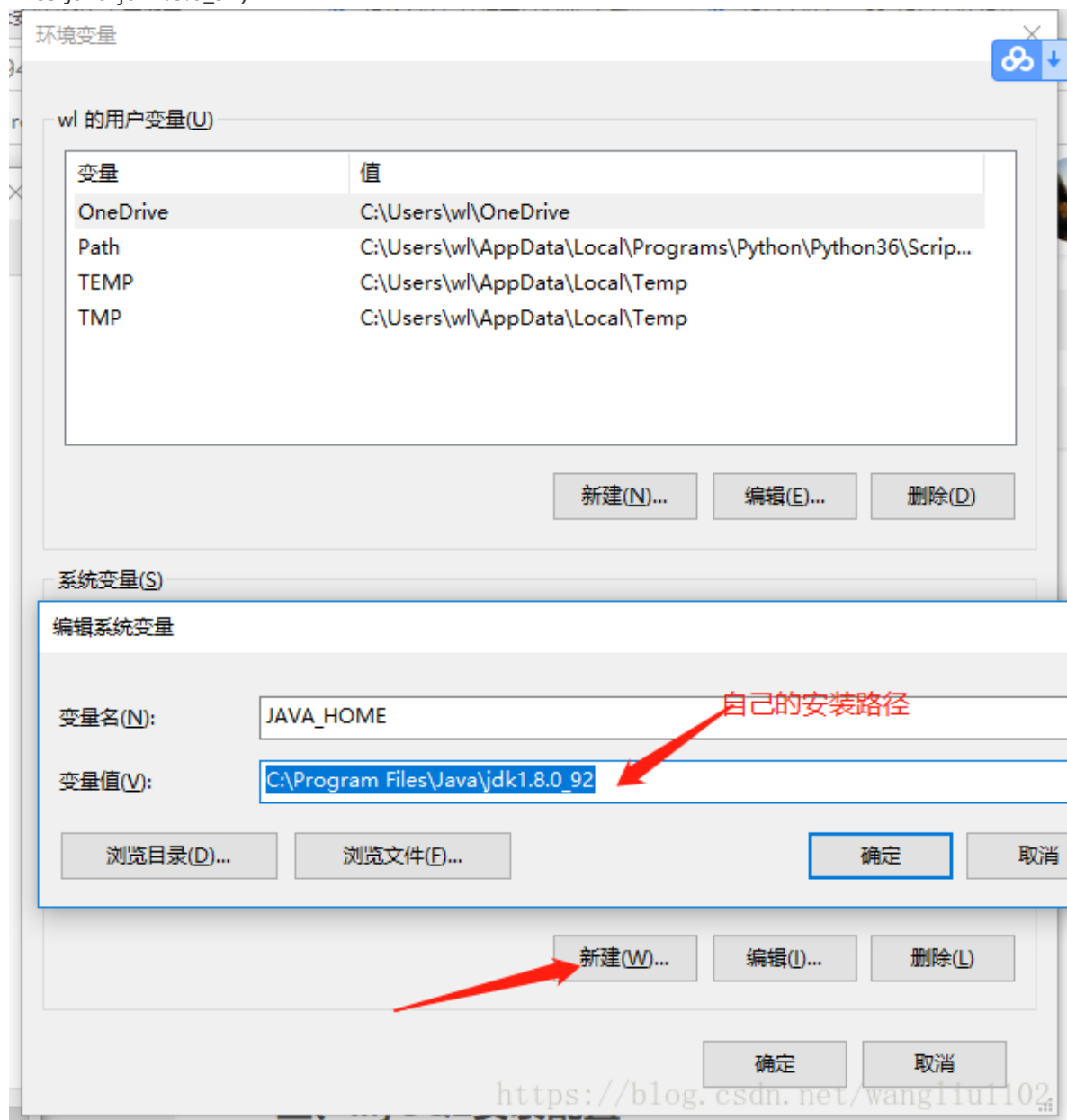
下载后点击运行它，一步一步安装至成功，不要怕就下一步下一步来。注意：64位和32位按照各自电脑多少位来安装哈！！



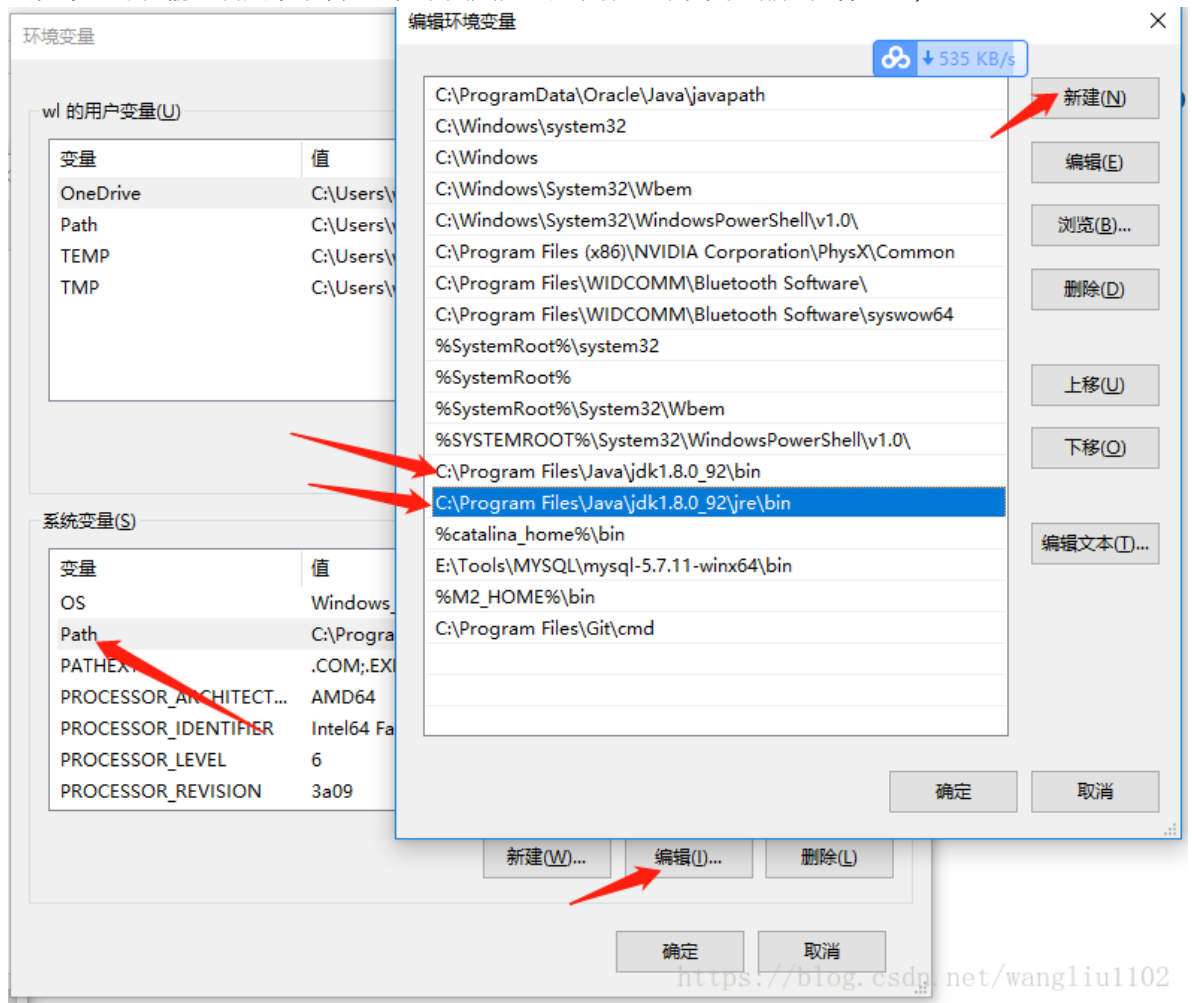
安装后并不能直接使用，要配置环境变量，步骤如下：1、打开计算机→属性→高级系统设置→环境变量



2、系统变量→新建 JAVA\_HOME 变量 变量值填写jdk的安装目录 (本人是 C:\Program Files\Java\jdk1.8.0\_92)



The screenshot shows the 'Edit Environment Variables' window in Windows. The 'Path' variable is selected in the 'System Variables' list. The 'Edit' button is highlighted with a red arrow. The 'Edit Environment Variables' window is open, showing a list of paths. The path 'C:\Program Files\Java\jdk1.8.0\_92\bin' is highlighted in blue. The 'New' button is also highlighted with a red arrow.



环境变量

wl 的用户变量(U)

变量	值
OneDrive	C:\Users\wl\OneDrive
Path	C:\Users\wl\AppData\Local\Programs\Python\Python36\Script...
TEMP	C:\Users\wl\AppData\Local\Temp
TMP	C:\Users\wl\AppData\Local\Temp

新建(N) 编辑(E) 删除(D)

编辑系统变量

系统变量(S)

变量	值
CATALINA_HOME	
CLASSPATH	;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar;D:\maven\apache-maven-3.3.9\lib\*.jar
ComSpec	C:\Windows\system32\cmd.exe
JAVA_HOME	C:\Program Files\Java\jdk-8.0.660\jre\bin
M2_HOME	D:\maven\apache-maven-3.3.9
NUMBER_OF_PROCESSORS	4
OS	Windows_NT

浏览目录(D)... 浏览文件(F)... 确定 取消

新建(W)... 编辑(I)... 删除(L)

确定 取消

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.16299.371]
(c) 2017 Microsoft Corporation. 保留所有权利。

C:\Users\wl>java -version
java version "1.8.0_92"
Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.92-b14, mixed mode)

C:\Users\wl>javac
用法: javac <options> <source files>
其中, 可能的选项包括:
-g 生成所有调试信息
-g:none 不生成任何调试信息
-g:{lines,vars,source} 只生成某些调试信息
-nowarn 不生成任何警告
-verbose 输出有关编译器正在执行的操作的消息
-deprecation 输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径> 指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 覆盖引导类文件的位置
-extdirs <目录> 覆盖所安装扩展的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-proc:{none,only} 控制是否执行注释处理和/或编译。
-processor <class1>[, <class2>... ] 要运行的注释处理程序的名称; 绕过默认的搜索进程
-processorpath <路径> 指定查找注释处理程序的位置
-parameters 生成元数据以用于方法参数的反射
-d <目录> 指定放置生成的类文件的位置
-s <目录> 指定放置生成的源文件的位置
-h <目录> 指定放置生成的本机标头文件的位置

```

<https://github.com/wangliu1102/StudyNotes/tree/master/Linux/%E5%AE%89%E8%A3%85>

## 六、开发体验—HelloWorld

### ●步骤:

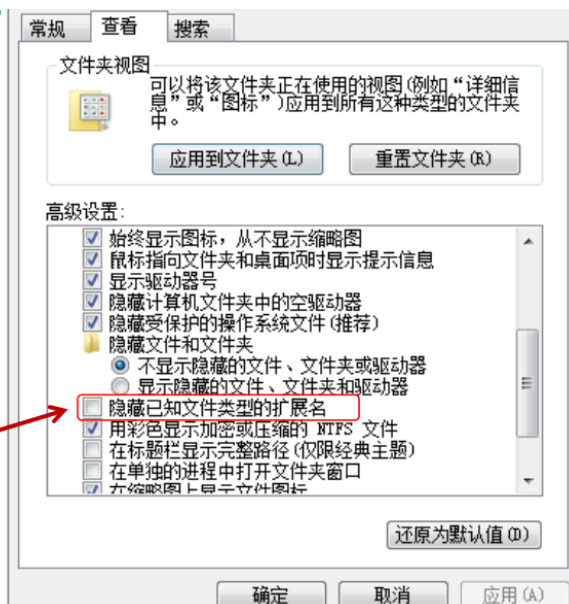
1. 将 Java 代码**编写**到扩展名为 .java 的文件中。
2. 通过 javac 命令对该 java 文件进行**编译**。
3. 通过 java 命令对生成的 class 文件进行**运行**。



### ● 步骤一：编写

- 选择最简单的编辑器：记事本。
- 敲入代码 `class Test{ }` 将文件保存成 `Test.java`，这个文件是存放java代码的文件，称为源文件。

取消勾选



### • 第一个Java程序

```
public class Test{  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```



## ● 步骤二：编译

```
D:\>javac Test.java
```

- 有了java源文件，通过编译器将其编译成JVM可以识别的字节码文件。
- 在该源文件目录下，通过javac编译工具对Test.java文件进行编译。
- 如果程序没有错误，没有任何提示，但在当前目录下会出现一个Test.class文件，该文件称为字节码文件，也是可以执行的java的程序。

## ● 步骤三：运行

- 有了可执行的java程序(Test.class字节码文件)
- 通过运行工具java.exe对字节码文件进行执行。
- 出现提示：缺少一个名称为main的方法。

```
D:\>java Test
错误: 在类 Test 中找不到主方法。请将主方法定义为:
    public static void main(String[] args)
```

- 因为一个程序的执行需要一个起始点或者入口，所以在Test类中的加入public static void main(String[] args){ }
- 对修改后的Test.java源文件需要重新编译，生成新的class文件后，再进行执行。
- 发现没有编译失败，但也没有任何效果，因为并没有告诉JVM要帮我们做什么事情，也就是没有可以具体执行的语句。
- 想要和JVM来个互动，只要在main方法中加入一句System.out.println("Hello World");因为程序进行改动，所以再重新编译，运行即可。

# 七、常见问题及解决方法

```
D:\>javac Test1.java
javac: 找不到文件: Test1.java
用法: javac <options> <source files>
-help 用于列出可能的选项
```

- 源文件名不存在或者写错
- 当前路径错误
- 后缀名隐藏问题

```
D:\>java Test1
错误: 找不到或无法加载主类 Test1
```

- 类文件名写错，尤其文件名与类名不一致时，要小心
- 类文件不在当前路径下，或者不在classpath指定路径下

```
D:\>javac Test.java
Test.java:1: 错误: 类Test1是公共的, 应在名为 Test1.java 的文件中声明
public class Test1{
      ^
1 个错误
```

➤ 声明为public的类应与文件名一致，否则编译失败

```
D:\>javac Test.java
Test.java:3: 错误: 需要';'
    System.out.println("hello")
                        ^
1 个错误
```

➤ 编译失败，注意错误出现的行数，再到源代码中指定位置改错

### 总结:

学习编程最容易犯的错是[语法错误](#)。Java要求你必须按照语法规则编写代码。如果你的程序违反了语法规则，例如：忘记了分号、大括号、引号，或者拼错了单词，java编译器都会报语法错误。[尝试着去看懂编译器会报告的错误信息。](#)

## 八、注释(Comment)

---

- 用于注解说明解释程序的文字就是注释。
- Java中的注释类型：
  - 单行注释
  - 多行注释
  - 文档注释 (java特有)
- [提高了代码的阅读性；调试程序的重要方法。](#)
- 注释是一个程序员必须要具有的良好编程习惯。
- 将自己的思想通过注释先整理出来，再用代码去体现

### 1、单/多行注释

---

- 单行注释

- 格式: `//注释文字`

- 多行注释

- 格式: `/* 注释文字 */`

- 注:

- 对于单行和多行注释, 被注释的文字, 不会被JVM (java虚拟机) 解释执行。

- 多行注释里面不允许有多行注释嵌套。

## 2、文档注释 (Java特有)

- 文档注释 (Java特有)

- 格式: `/**`

- `@author` 指定java程序的作者

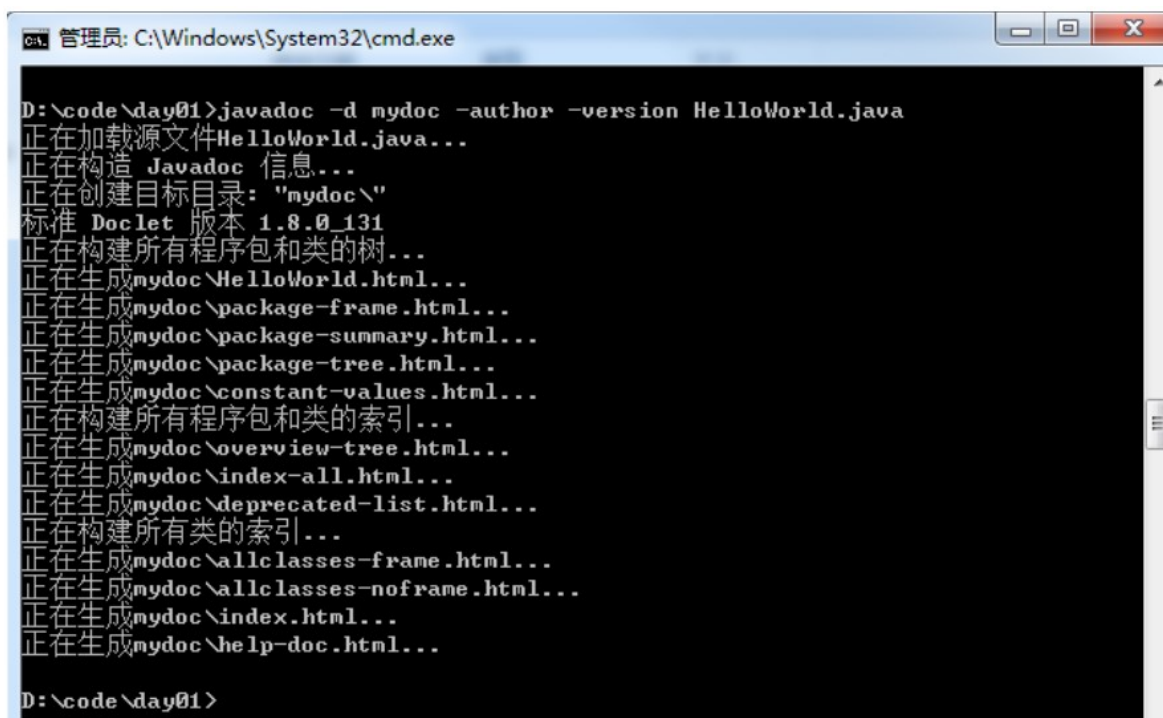
- `@version` 指定源文件的版本

- `*/`

- 注释内容可以被JDK提供的工具 `javadoc` 所解析, 生成一套以网页文件形式体现的该程序的说明文档。

- 操作方式

```
D:\javase\code\unit1>javadoc -d mydoc -author -version HelloWorld.java
```



```
C:\Windows\System32\cmd.exe
D:\code\day01>javadoc -d mydoc -author -version HelloWorld.java
正在加载源文件HelloWorld.java...
正在构造 Javadoc 信息...
正在创建目标目录: "mydoc\"
标准 Doclet 版本 1.8.0_131
正在构建所有程序包和类的树...
正在生成mydoc\HelloWorld.html...
正在生成mydoc\package-frame.html...
正在生成mydoc\package-summary.html...
正在生成mydoc\package-tree.html...
正在生成mydoc\constant-values.html...
正在构建所有程序包和类的索引...
正在生成mydoc\overview-tree.html...
正在生成mydoc\index-all.html...
正在生成mydoc\deprecated-list.html...
正在构建所有类的索引...
正在生成mydoc\allclasses-frame.html...
正在生成mydoc\allclasses-noframe.html...
正在生成mydoc\index.html...
正在生成mydoc\help-doc.html...
D:\code\day01>
```

## 3、小结第一个程序

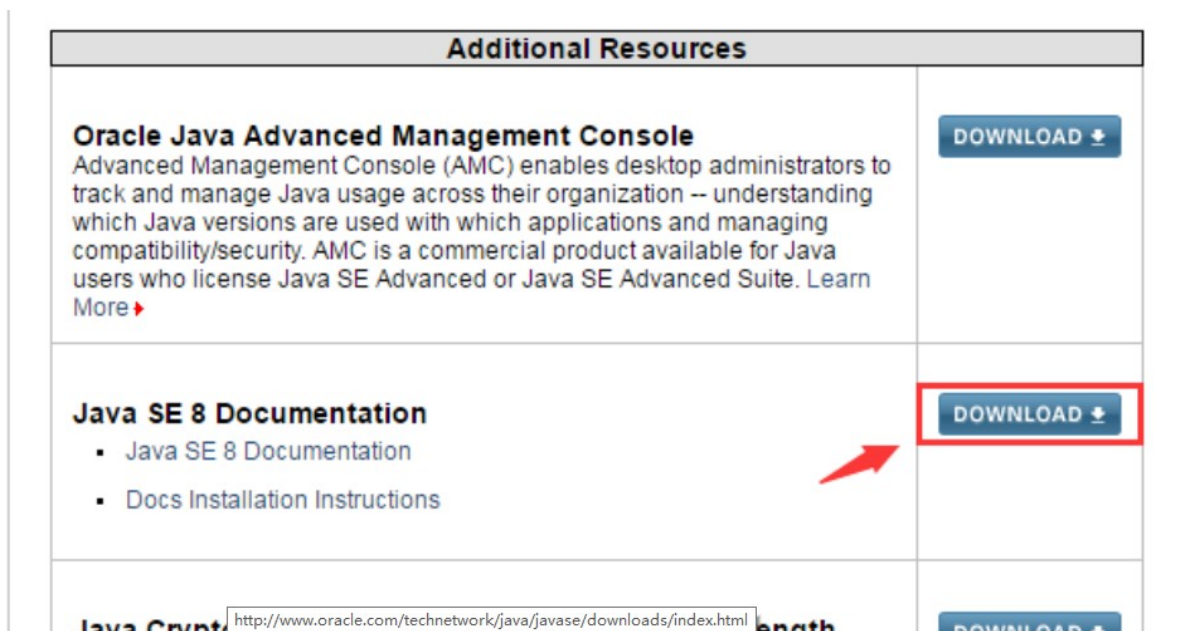
- Java源文件以“java”为扩展名。源文件的基本组成部分是类（class），如本例中的HelloWorld类。
- Java应用程序的执行入口是main()方法。它有固定的书写格式：  
**public static void main(String[] args) {...}**
- Java语言严格区分大小写。
- Java方法由一条条语句构成，每个语句以“;”结束。
- 大括号都是成对出现的，缺一不可。
- 一个源文件中最多只能有一个public类。其它类的个数不限，如果源文件包含一个public类，则文件名必须按该类名命名。

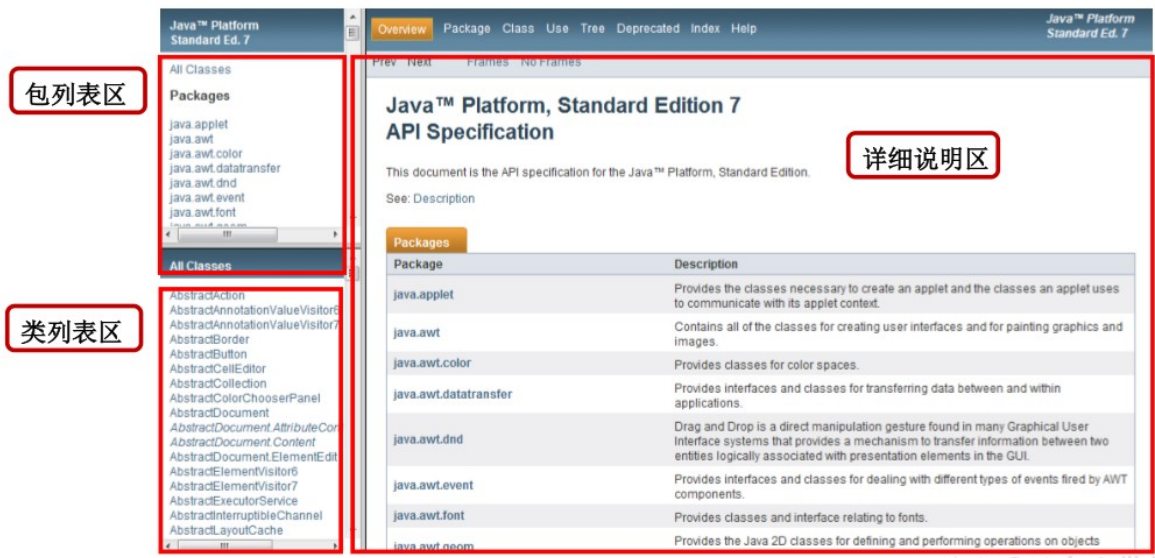
## 九、Java API文档

- API（Application Programming Interface,应用程序编程接口）是Java提供的基本编程接口。
- Java语言提供了大量的基础类，因此Oracle也为这些基础类提供了相应的API文档，用于告诉开发者如何使用这些类，以及这些类里包含的方法。
- 下载API:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

➤ Additional Resources-Java SE 8 Documentation下载。





## 十、良好的编程风格

- 正确的注释和注释风格
  - 使用文档注释来注释整个类或整个方法。
  - 如果注释方法中的某一个步骤，使用单行或多行注释。
- 正确的缩进和空白
  - 使用一次tab操作，实现缩进
  - 运算符两边习惯性各加一个空格。比如：2 + 4 \* 5。
- 块的风格
  - Java API 源代码选择了行尾风格

```
public class Test {
    public static void main(String[] args){
        System.out.println("Block Style!");
    }
}
```

行尾风格

```
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Block Style!");
    }
}
```

次行风格

让天下没有难学的

## 十一、常用的Java开发工具



## ● 文本编辑工具：

- 记事本
- UltraEdit
- EditPlus
- TextPad
- NotePad

## ● Java集成开发环境（IDE）：

- JBuilder
- NetBeans
- Eclipse
- MyEclipse
- IntelliJ IDEA

## 十二、每日练习

---

### 1、JDK,JRE,JVM三者之间的关系，以及JDK、JRE包含的主要结构有哪些？

---

JDK = JRE + Java的开发工具 (javac.exe,java.exe,javadoc.exe)

JRD = JVM + Java核心类库

### 2、java语言的特点是什么？

---

- 面向对象性：两个基本概念：类、对象；三大特性：封装、继承、多态
- 健壮性：吸收了C/C++语言的优点，但去掉了其影响程序健壮性的部分（如指针、内存的申请与释放等），提供了一个相对安全的内存管理和访问机制

- 跨平台性：通过Java语言编写的应用程序在不同的系统平台上都可以运行。“Write once , Run Anywhere”

### 3、一个".java"源文件中是否可以包括多个类（不是内部类）？有什么限制？

---

可以。但最多只有一个类名声明为public，与文件名相同。

### 4、Something类的文件名叫OtherThing.java

---

```
class Something {  
    public static void main(String[] something_to_do) {  
        System.out.println("Do something ...");  
    }  
}
```

正确。从来没有人说过Java的class名字必须和其文件名相同。但public class的名字必须和文件名相同。

### 5、GC是什么？为什么要有GC

---

GC是垃圾收集的意思（Garbage Collection）。

内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java提供的GC功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java语言没有提供释放已分配内存的显示操作方法。

### 6、垃圾回收器的基本原理是什么？垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收

---

对于GC来说，当程序员创建对象时，GC就开始监控这个对象的地址、大小以及使用情况。通常，GC采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是"可达的"，哪些对象是"不可达的"。当GC确定一些对象为"不可达"时，GC就有责任回收这些内存空间。

可以。

程序员可以手动执行System.gc()，通知GC运行，但是Java语言规范并不保证GC一定会执行。