1. Java语言概述

山西优逸客科技有限公司官网

*http://www.sxuek.com/*

|  |  |  |
| --- | --- | --- |
| 编辑时间 | 编辑版本 | 编辑人 |
| 2019-7-12 | V2.0 | 优逸客大数据实训组 |
|  |  |  |

# 第一章 计算机知识储备：

## 1.1 概述：

计算机包括硬件(hardware)和软件(software)两部分。

硬件包括计算机中可以看得见的物理部分；

软件提供看不见的指令。这些指令控制硬件并且使得硬件完成特定的任务；

程序设计：

定义：创建（或开发）软件。软件包含了指令，告诉计算机做什么。

应用场景：

软件遍布我们周围。除了个人计算机，飞机、汽车、手机甚至烤面包机中，同样运行着软件。

程序设计语言：

软件开发人员在称为程序设计语言的强大工具的帮助下创建软件。

如何选择该学习哪种程序设计语言？

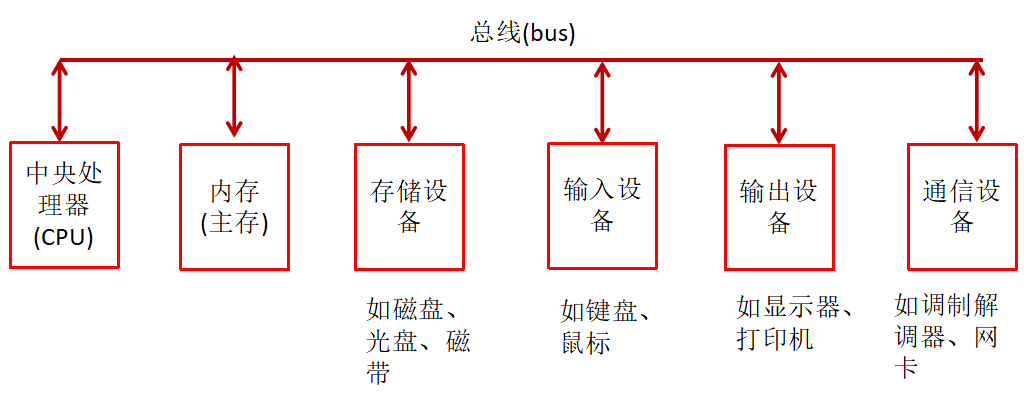
1）程序设计语言有很多种，每种语言都是为了实现某个特定的目的而发明的。

2）你会困惑哪种语言是最好的。事实上，没有“最好”的语言。每种语言都有它的长处和短处。

3）经验丰富的程序员知道各种语言擅长的应用场景，因此，会尽可能的掌握各种不同的程序设计语言。

4）如果你掌握了一种编程语言，应该会更容易上手其它的编程语言。

## 1.2 计算机硬件介绍：



在个人计算机上，总线搭建在主板上，主板是一个连接计算机各个部分的电路板。

中央处理器：



中央处理器(Central Processing Unit,CPU)是计算机的大脑。它从内存中获取指令，然后执行这些指令。

包括：控制单元(control unit)和算术/逻辑单元(arithmetic/login unit)。

控制单元：用于控制和协调其他组件的动作。

算术/逻辑单元：用于完成数值运算(+、-、\*、/)和逻辑运算(比较)。

每台计算机都有一个内部时钟，该时钟以固定速度发射电子脉冲。时钟速度越快，在给定的时间段内执行的指令就越多。速度的计量单位是赫兹(Hz)，1Hz相当于每秒1个脉冲。随着CPU速度不断提高，目前以千兆赫(GHz)来表述。

最初一个CPU只有一个核(core)。核是处理器中实现指令读取和执行的部分。一个多核CPU是一个具有两个或者更多独立核的组件。可提高CPU的处理能力。

现在的CPU都是构建在一块小小的硅半导体芯片上，这块芯片上包含数百万称为晶体管的小电路开关上，用于处理信息。

比特(bit)和字节(byte)：

在讨论内存前，先清楚数据时如何存储在计算机中的。

计算机就是一系列的电路开关。每个开关存在两种状态：关(off)和开(on)。如果电路是开的，它的值是1。如果电路是关的，它的值是0。

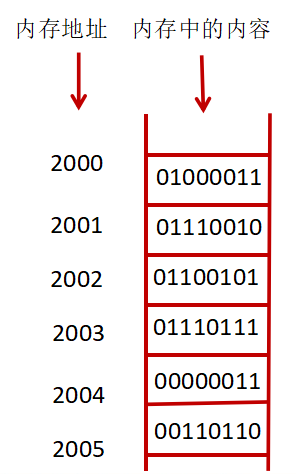
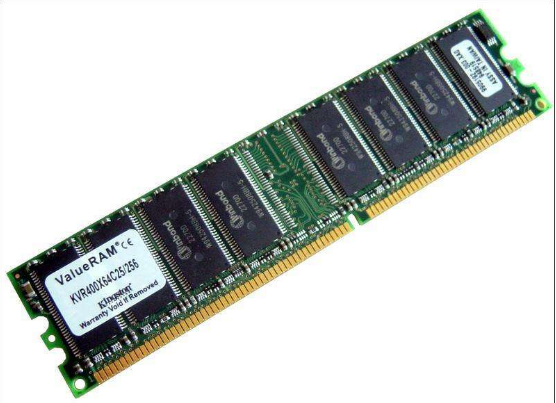
一个0或者一个1存储为一个比特(bit)[2进制位]。

计算机中字节(byte)是最小的存储单元。每个字节由8个比特构成。

计算机的存储能力是以字节和多字节来衡量的。如下：

|  |
| --- |
| *千字节(kilobyte,KB) = 1024B*  *兆字节(megabyte,MB) = 1024KB*  *千兆字节(gigabyte,GB) = 1024MB*  *万亿字节(terabyte,TB) = 1024GB* |

内存：



内存(也叫 Random-Access Memory,RAM)：由一个有序的字节序列组成，用于存储程序及程序需要的数据。

一个程序和它的数据在被CPU执行前必须移到计算机的内存中。

每个字节都有一个唯一的地址。使用这个地址确定字节的位置，以便于存储和获取数据。

一个计算机具有的RAM越多，它的运行速度越快，但是此规律是有限制的。

内存与CPU一样，也构建在表面嵌有数百万晶体管的硅半导体芯片上。但内存芯片更简单、更低速、更便宜。

存储设备：

内存中的信息在断电时会丢失。那我们可以考虑将程序和数据永久的保存在存储设备上。当计算机确实需要这些数据时，再移入内存，因为从内存中读取比从存储设备读取要快得多。

存储设备主要有以下三种：

1. 磁盘驱动器：

每台计算机至少有一个硬盘驱动器。硬盘(hard disk)用于永久的保存数据和程序。

1. 光盘驱动器(CD和DVD)：

CD的容量可达700MB。DVD的容量可达4.7GB。

3）USB闪存驱动器：

USB: Universal Serial Bus，通用串行总线。

可以使用USB将打印机、数码相机、鼠标、外部硬盘驱动器连接到计算机上。

USB闪存驱动器很小，可用于存储和传输数据的设备。

输入和输出设备：

常见的输入设备：键盘（keyboard）和鼠标（mouse）

常见的输出设备：显示器（monitor）和打印机（printer）

显示器屏幕分辨率：是指显示设备水平和垂直方向上显示的像素数。分辨率可以手工设置。分辨率越高，图像越锐化、越清晰。

通信设备：

计算机可以通过通信设备进行联网。

常见的设备有：

1. 拨号调制解调器：

使用的是电话线，传输速度可达56 000bps(bps:每秒比特)

1. DSL（数字用户线）：

使用的也是电话线，但传输速度叫上面的快20倍

1. 电缆调制解调器（猫）：

利用有线电视电缆进行数据传输，通常速度比DSL快。

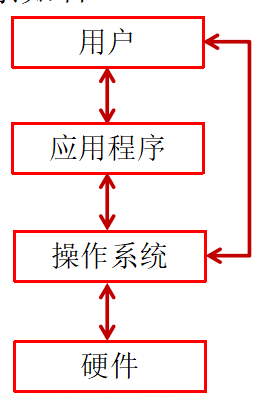
1. 网络接口卡（NIC）：

将计算机接入局域网（LAN）的设备。局域网通常用于大学、商业组织和政府组织。速度甚至可达1000Mbps

1. 无线网络：

在家庭、商业和学校中极其常见。计算机可通过无线适配器连接到局域网或internet上。

操作系统：



操作系统(Operating System)是运行在计算机上的最重要的程序，它可以管理和控制计算机的活动。

硬件、操作系统、应用软件和用户之间的关系。

操作系统的主要任务：

1）控制和监视系统的活动；

2）分配和调配系统资源；

3）调度操作；

万维网：

万维网（World Wide Web,www）是从世界上任何地方的Internet都可以访问的电子信息宝库。Internet作为万维网的基础架构已经问世四十多年。丰富多彩的万维网和设计精良的Web浏览器是Internet流行的主要原因。

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。

## 1.3 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。

|  |
| --- |
| *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*  *echo javase > 1.doc ： 创建一个文件，同时写入内容*  *del : 删除文件*  *exit : 退出 dos 命令行*  *常用快捷键*  *← →：移动光标*  *↑ ↓：调阅历史操作命令*  *Delete和Backspace：删除字符* |

## 1.5 计算机编程语言介绍：

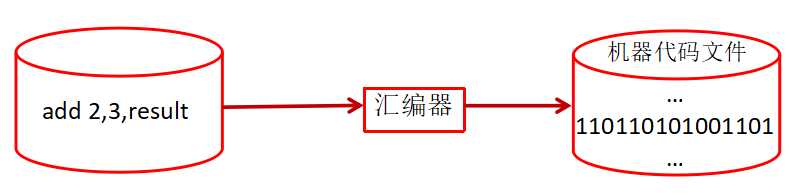
什么是计算机语言？

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。

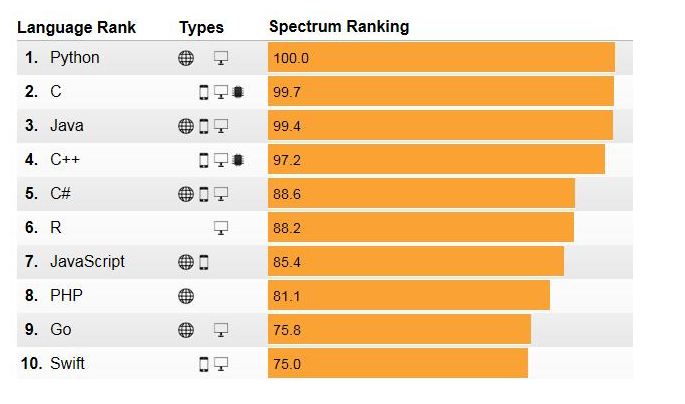
第二代语言：汇编语言。使用助记符表示一条机器指令。



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；

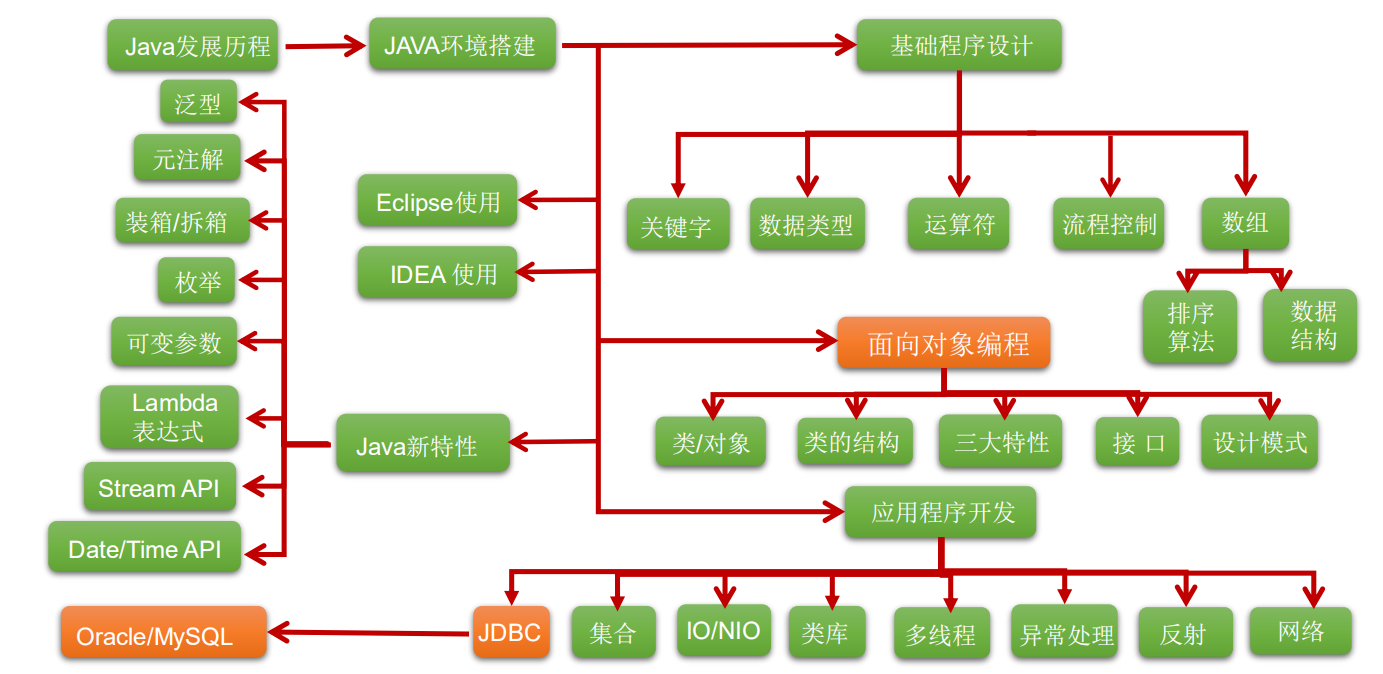


从java语言的市场需求来看：



# 第二章 Java语言概述：





Java基础知识图解

## 2.1 Java基础课程概述：

第一部分：编程语言核心结构

主要知识点：变量、基本语法、分值、循环、数组、…

第二部分：Java面向对象的核心逻辑

主要知识点：OOP、封装、继承、多态、接口、…

第三部分：开发Java SE高级应用程序

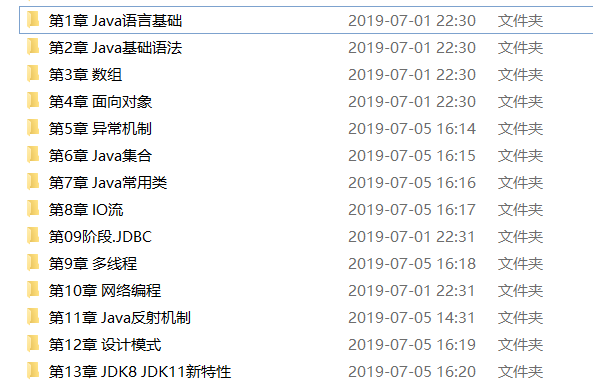
主要知识点：异常、集合、I/O、多线程、反射机制、网络编程、……

第四部分：实训项目

项目一：技术点；

项目二：微项目（市场需求调研）；

## 2.2 Java基础课程体系：



## 2.3 本章内容：

1、程序

2、计算机语言

2、JAVA语言的介绍

3、JAVA程序的运行机制和执行过程

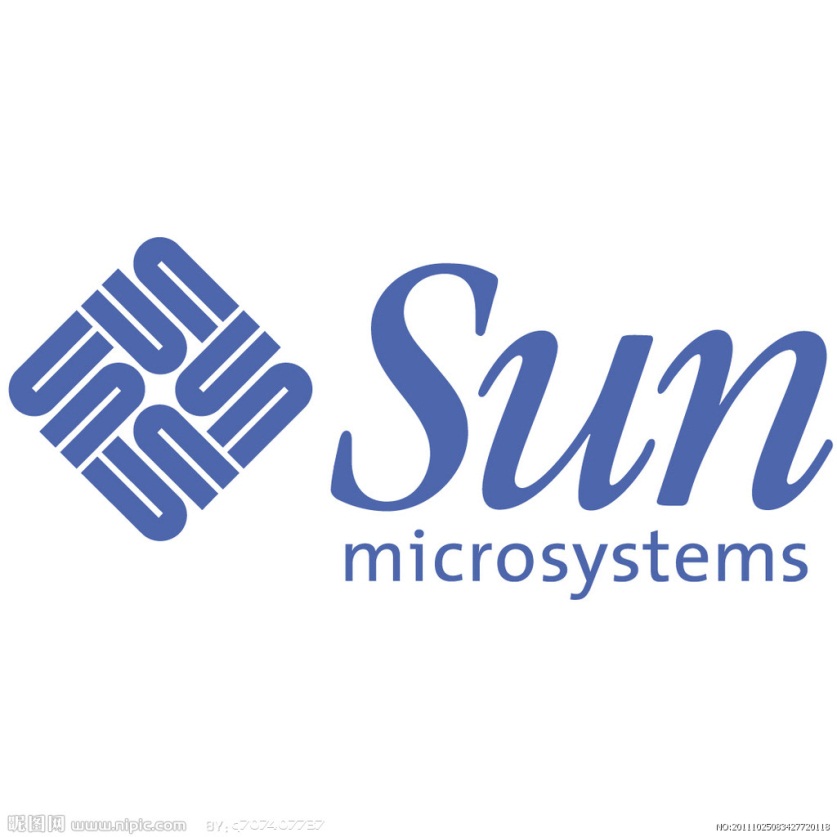
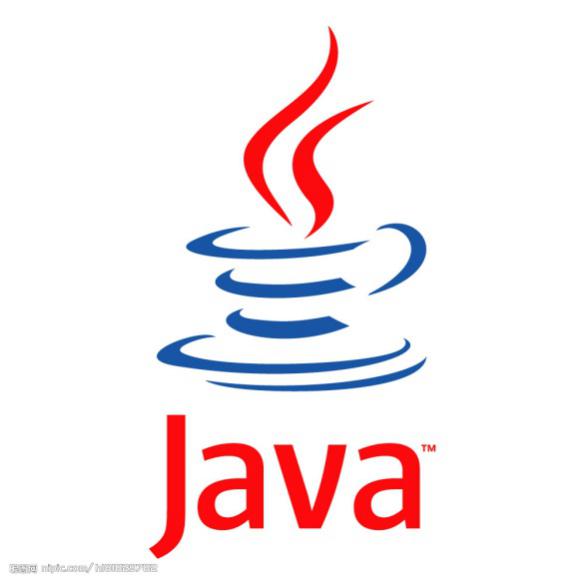
4、开发JAVA程序的步骤

5、开发JAVA第一个程序

6、常见注释和API

7、常见的dos命令【补充】

## 2.4 Java语言概述：





是SUN(Stanford University Network，斯坦福大学网络公司)1995年推出的一门高级编程语言。

* 最初命名为Oak (橡树)
* 最初的目的：与家电一起使用
* 1994年，开发组意识到Oak 非常适合于互联网
* 1995年，SUN发布JDK 1.0
* 1996年，发布正式版
* 1998年，JDK1.2，后续JDK1.3， 1.4，1.5（更名为Java5.0）
* 最新版本为 JDK 11。

是一种面向Internet的编程语言。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。

### 2.4.1 Java技术体系平台：



### 2.4.2 Java在各领域中的应用：

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。

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。

### 2.4.3 java语言的诞生历史：



java之父Jgosling（[詹姆斯·高斯林](https://baike.baidu.com/item/%E8%A9%B9%E5%A7%86%E6%96%AF%C2%B7%E9%AB%98%E6%96%AF%E6%9E%97/9902700" \t "https://baike.baidu.com/item/Java%E4%B9%8B%E7%88%B6/_blank)）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。

### 2.4.4 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。

## 2.5 Java语言运行机制及运行过程：

### 2.5.1 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

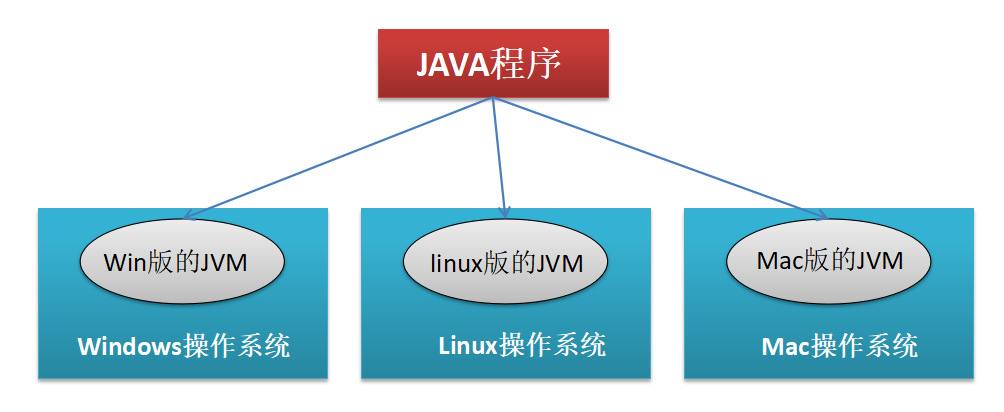
吸收了C/C++语言的优点，但去掉了其影响程序健壮性的部分（如指针、内存的申请与释放等），提供了一个相对安全的内存管理和访问机制.

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 Java Virtual Machine) 即可。由JVM来负责Java程序在该系统中的运行。

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。



### 2.5.2 Java两种核心机制：

Java虚拟机（Java Virtal Machine）

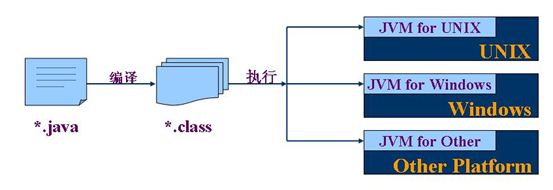
垃圾收集机制（Garbage Collection）

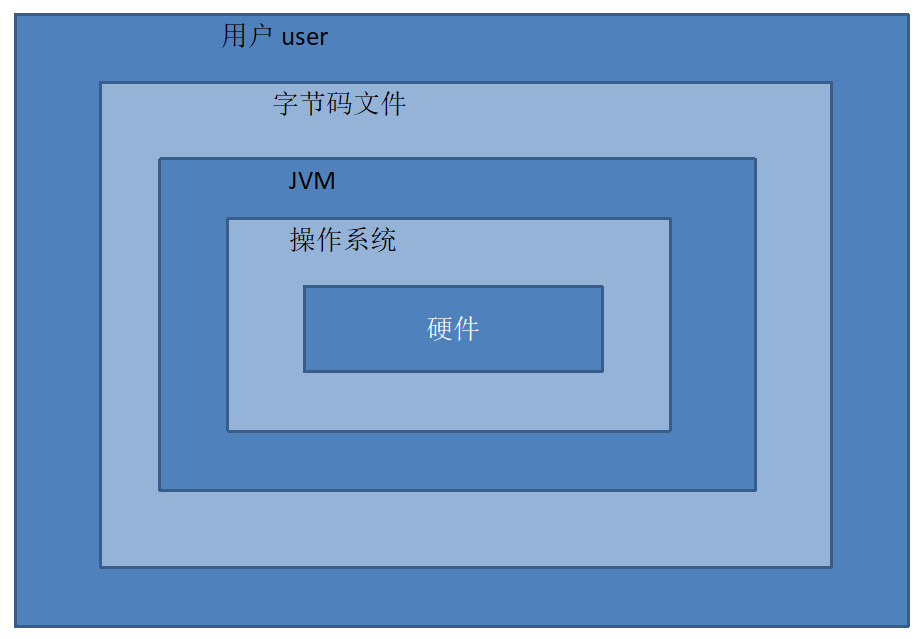
核心机制—Java虚拟机：

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。

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”





核心机制—垃圾回收：

不再使用的内存空间自动回收—— 垃圾回收。

在C/C++等语言中，由程序员负责回收无用内存。

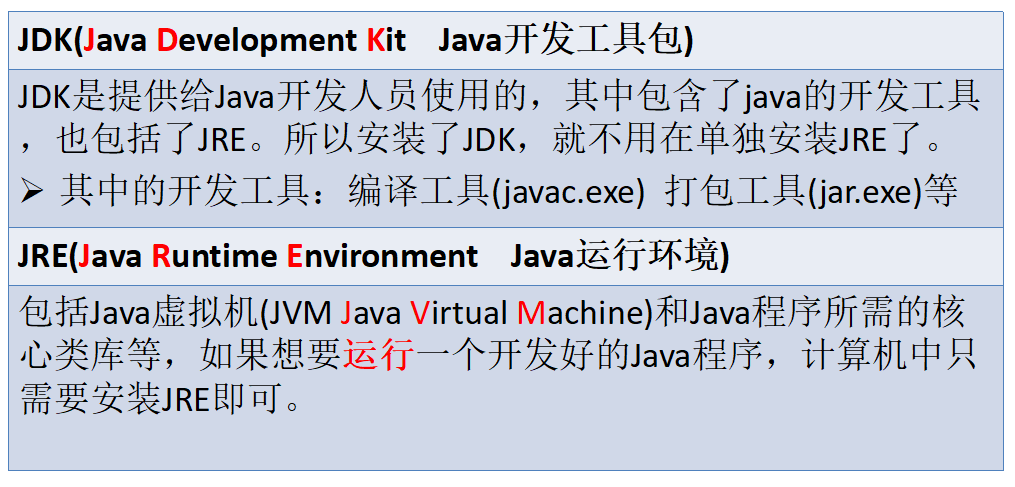
Java 语言消除了程序员回收无用内存空间的责任：它提供一种系统级线程跟踪存储空间的分配情况。并在JVM空闲时，检查并释放那些可被释放的存储空间。

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。

## 2.6 Java语言的环境搭建：

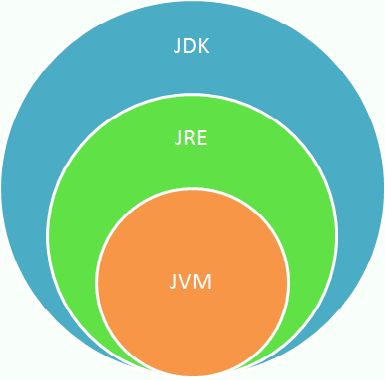
* 明确什么是JDK, JRE
* 下载 JDK
* 安装 JDK
* 配置环境变量
* 验证是否成功：javac java
* 选择合适的文本编辑器或 IDE 开发

### 2.6.1 什么是JDK，JRE：



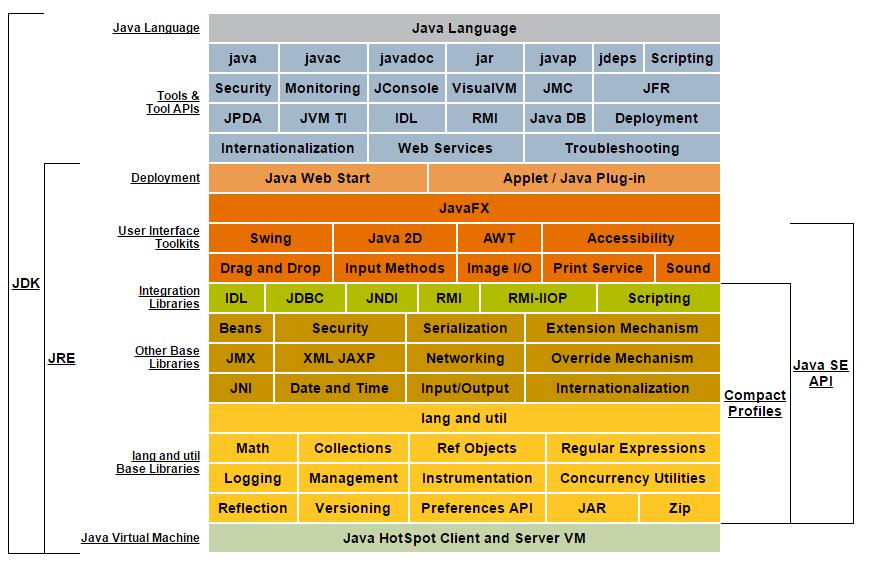
简单而言，使用JDK的开发工具完成的java程序，交给JRE去运行。

### 2.6.2 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



### 2.6.3 下载、安装JDK：

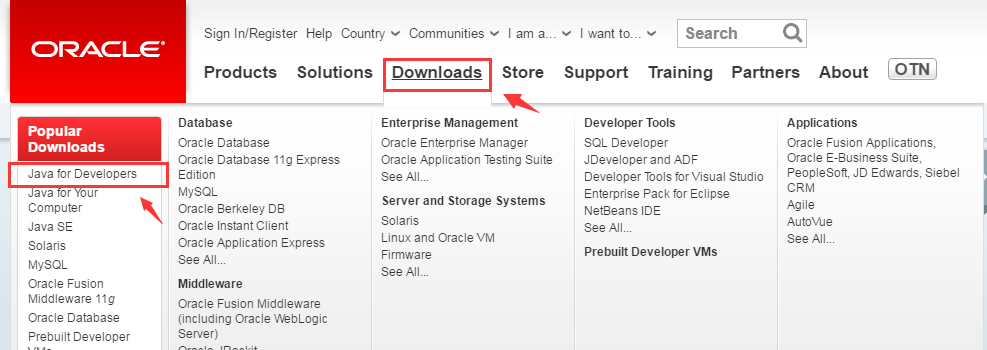
|  |
| --- |
| 官方网址：  *www.oracle.com*  *java.sun.com*  安装JDK  傻瓜式安装，下一步即可。  建议：安装路径不要有中文或者特殊符号如空格等。  当提示安装 JRE 时，可以选择不安装。 |

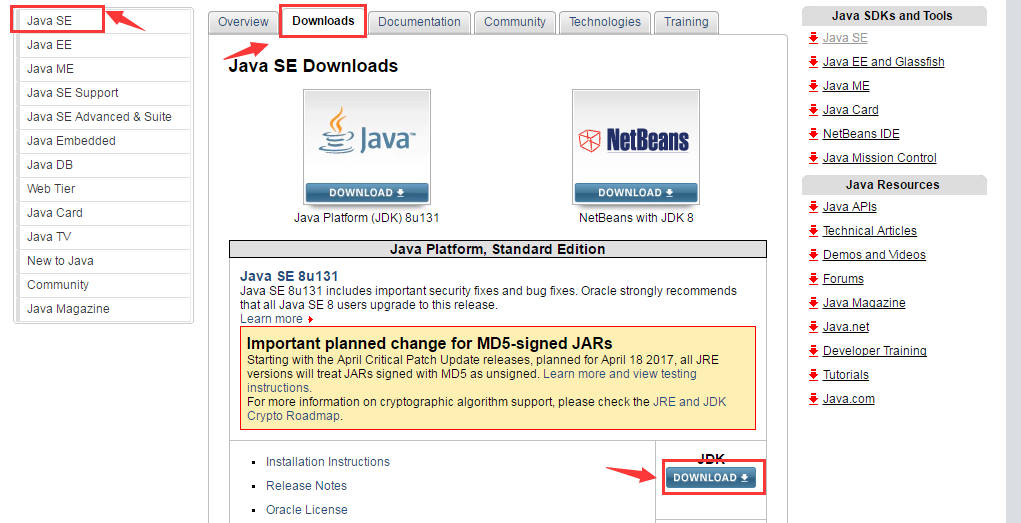
**JDK 8安装及配置：**

**一、安装环节：**

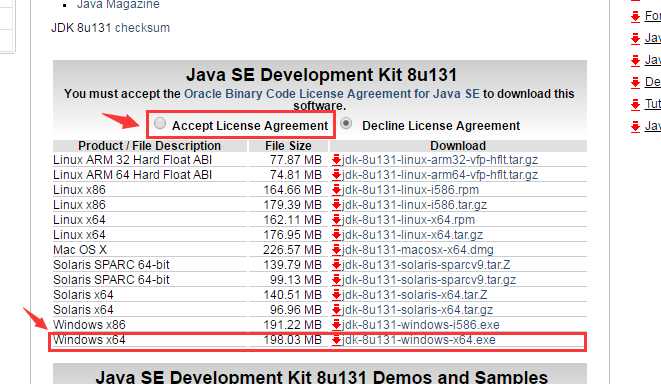
1. 打开网页

<http://www.oracle.com>





下载对应平台的合适的JDK版本。



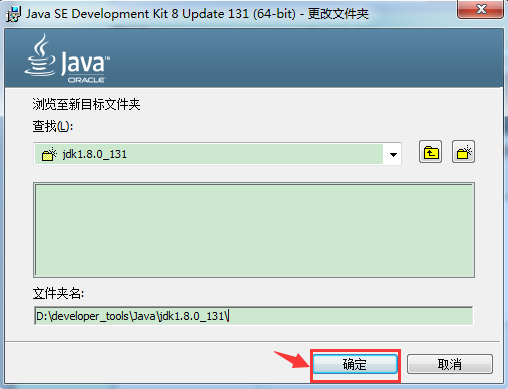
1. 双击下载的exe,如jdk-8u131-windows-x64.exe。
2. 进入安装向导。



1. 下一步,更改安装路径，选择安装所有组件



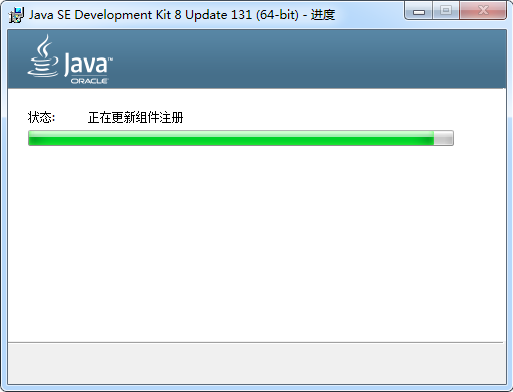
更改为D:\developer\_tools\Java\jdk1.8.0\_131\



点击确定



1. 下一步，开始安装



1. 安装jre，可以更改jre安装路径。(过程同上述安装目录的选择)

注意：如果提示需要将jre安装在一个空目录下，那自己创建一个目录即可。



1. 点击下一步，开始安装



1. 结束安装

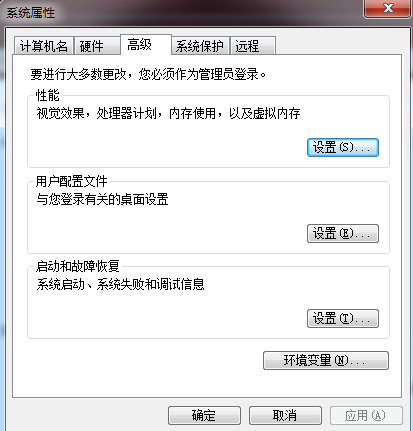


**二、安装完毕，下面是配置环节：**

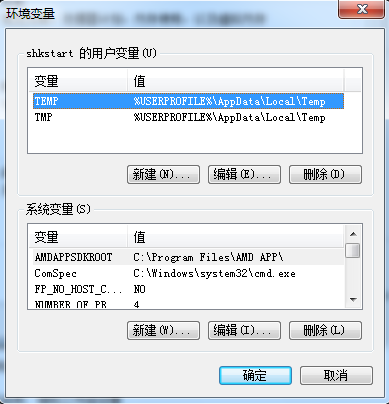
1. 选中桌面”我的电脑”-右键选择属性，选择高级系统设置。

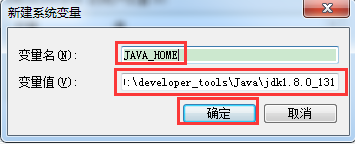


1. 点击环境变量：

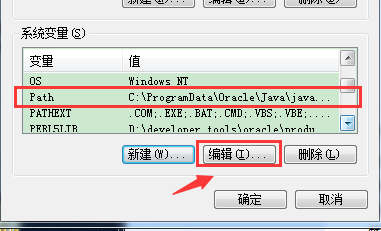


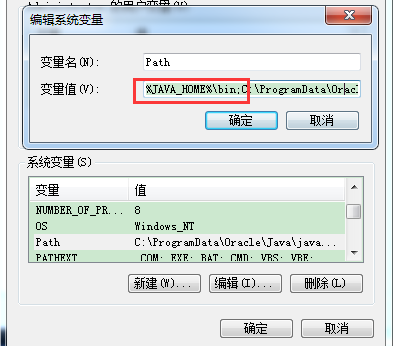
1. 新建一项系统变量“JAVA\_HOME”,值为jdk的安装路径。





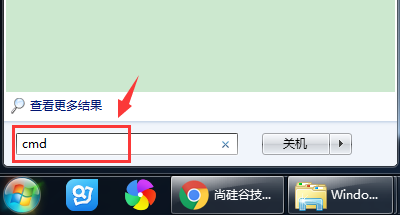
1. 配置系统变量:双击系统变量的path，在变量值最前端添加**%JAVA\_HOME%\bin;** 然后确定-确定

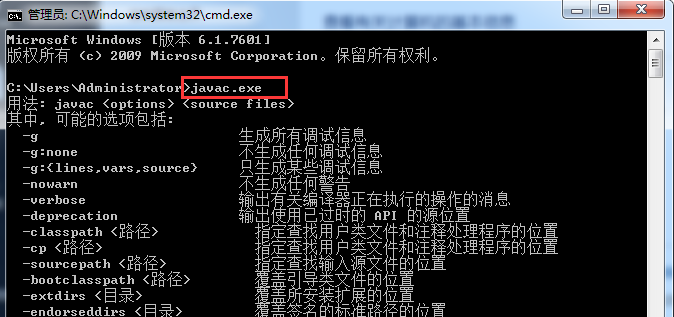




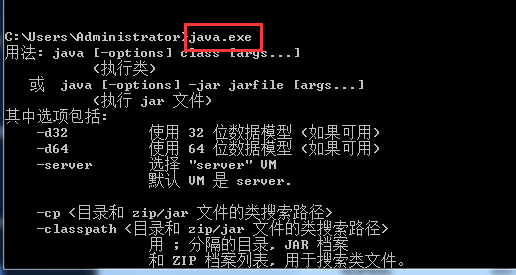
1. 检验是否配置成功：通过运行cmd指令，进入命令行窗口。

输入：javac.exe

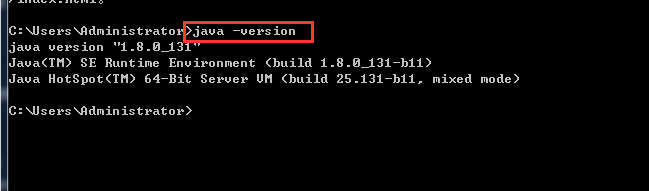




1. 检验java.exe命令



1. 获取当前安装的jdk的版本信息



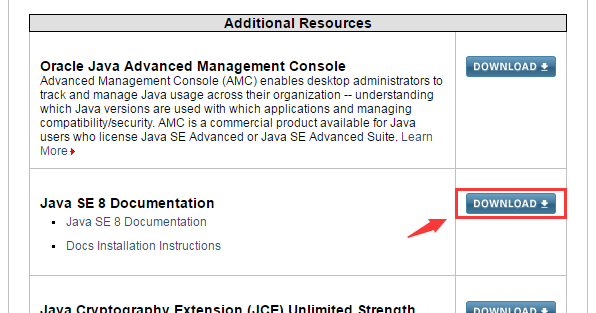
**三、配置完毕，下面是下载API：**

同时学习java少不了API（Application Programming Interface）文档。

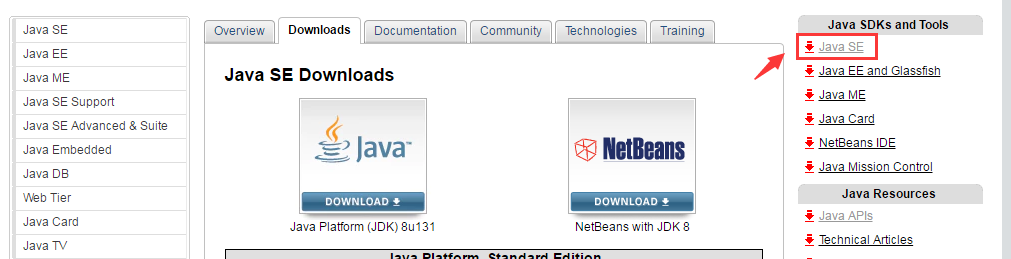
下载地址：

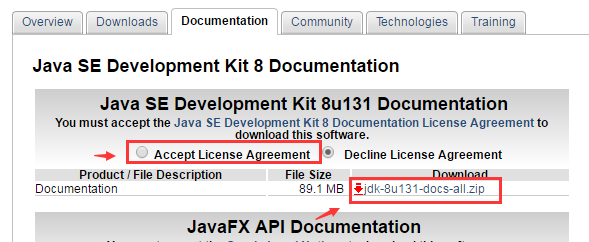
*<http://www.oracle.com/technetwork/java/javase/downloads/index.html>*

将页面上的滚动条向下滚动，找到”Additional Resources”部分。如下：



或者通过主页面，找到链接地址进来：





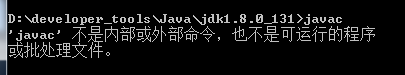
### 2.6.4 配置环境变量 path：

path：windows系统执行命令时要搜寻的路径。

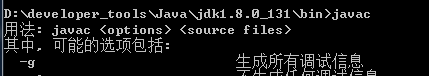
classpath：java 在编译和运行时要找的 class 所在的路径。



在dos命令行中敲入javac，出现错误提示：



错误原因：当前执行的程序在当前目录下如果不存在，windows系统会在系统中已有的一个名为path的环境变量指定的目录中查找。如果仍未找到，会出现以上的错误提示。所以进入到 jdk安装路径\bin目录下，执行javac，会看到javac参数提示信息。



每次执行 java 的工具都要进入到bin目录下，是非常麻烦的。可不可以在任何目录下都可以执行java的工具呢？

根据windows系统在查找可执行程序的原理，可以将java工具所在路径定义到 path 环境变量中，让系统帮我们去找运行执行的程序。

配置方法：

我的电脑--属性--高级系统设置--环境变量

编辑 path 环境变量，在变量值开始处加上java工具所在目录，后面用 “ ; ”和其他值分隔开即可。

打开DOS命令行，任意目录下敲入javac。如果出现javac 的参数信息，配置成功。

临时配置方式：通过dos命令中set命令完成

set ：用于查看本机的所有环境变量的信息。

set 变量名 ：查看具体一个环境变量的值。

set 变量名=：清空一个环境变量的值。

set 变量名=具体值 ：给指定变量定义具体值。

想要在原有环境变量值基础上添加新值呢？

首先，通过%变量名%操作符获取到原有环境变量的值。

然后加上新值后再定义给该变量名即可

举例：给path环境变量加入新值

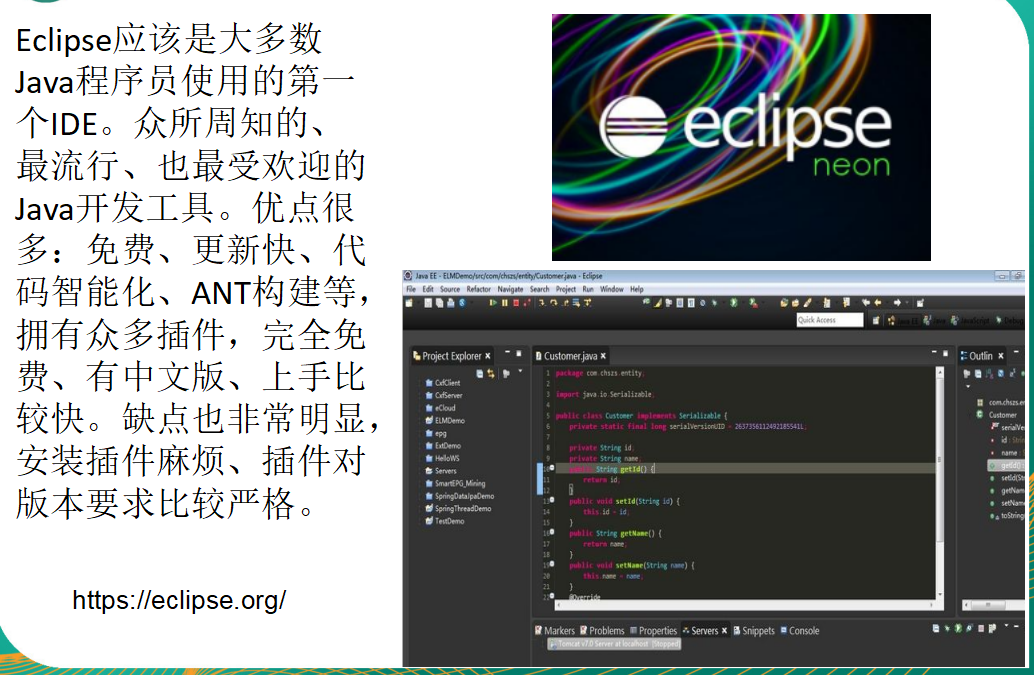
set path=新值;%path%

注：这种配置方式只在当前dos窗口有效。窗口关闭，配置消失。

### 2.6.5 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)：*  *IntelliJ IDEA Eclipse MyEclipse NetBeans* |







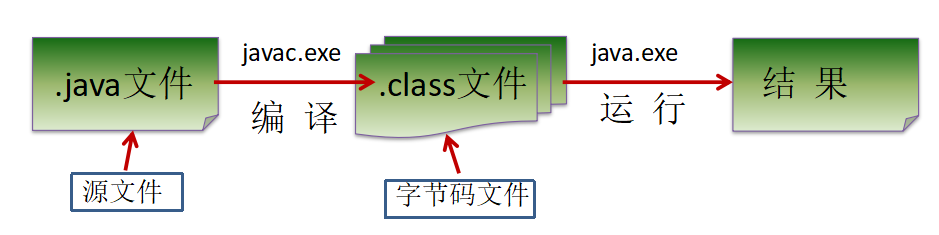
## 2.7开发体验 — HelloWorld：

### 2.7.1 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。

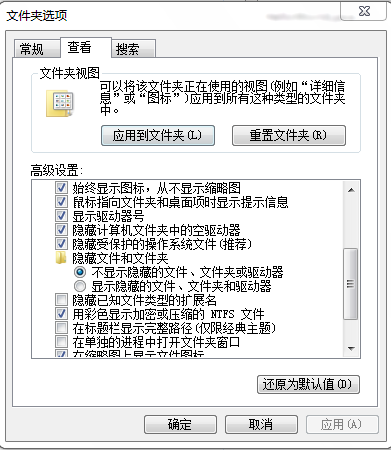


步骤一：编写

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。



|  |
| --- |
| *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



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。



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。

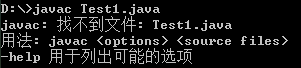
### 2.7.2 小结第一个程序：

Java源文件以“java”为扩展名。源文件的基本组成部分是类（class），如本类中的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。

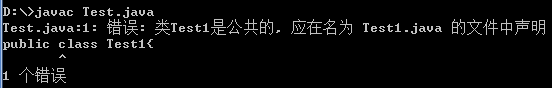
### 2.7.3 常见问题及解决方法：



源文件名不存在或者写错，或者当前路径错误。



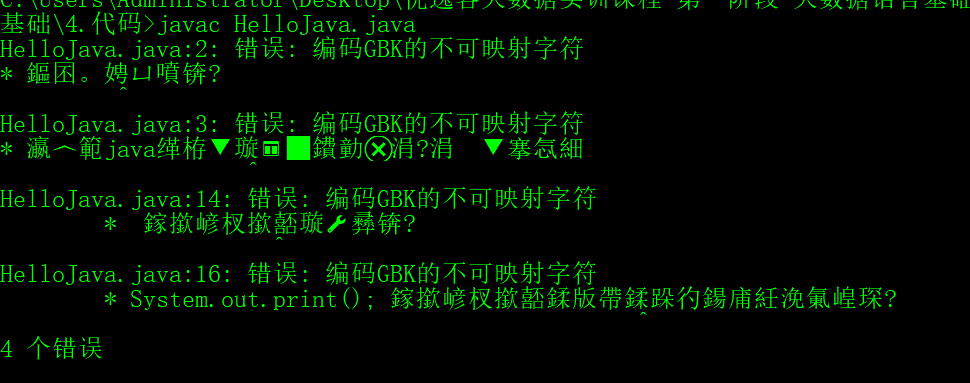
类文件名写错，或者类文件不在当前路径下，或者不在classpath指定路径下。



声明为public的主类应与文件名一致，否知编译失败



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；





Windows系统默认编码集是GBK(ANSI) , 而编辑器中为UTF-8 ，所以会报编码错误，需要统一成GBK(ANSI) 即可；

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。

## 2.8 注释(comment)

用于注解说明解释程序的文字就是注释。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。

### 2.8.1 Java中的注释类型：

1）单行注释：格式： //注释文字

2）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。

3）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

\*@param 方法的参数说明信息

\*/

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：

|  |
| --- |
| Javadoc -d 自定义文档名 -author -version HelloJava |



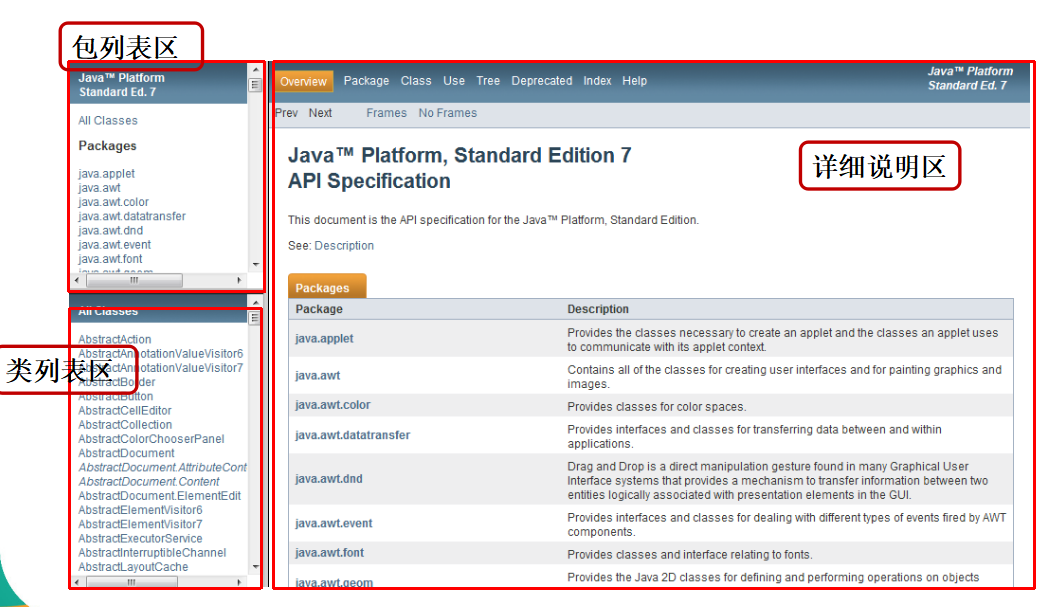
## 2.9 Java API文档：

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。



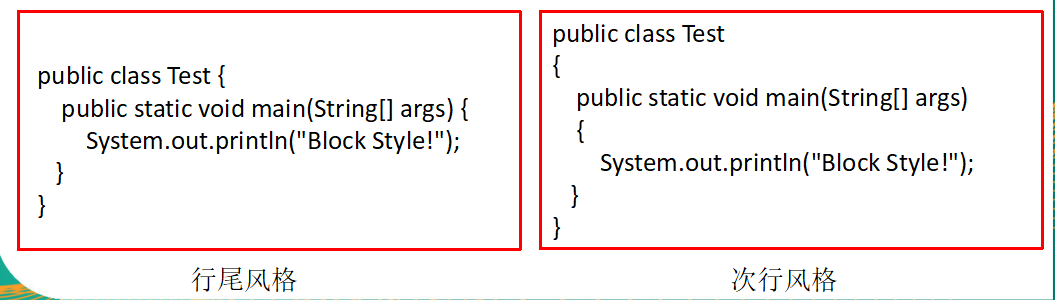
## 2.10 良好的编程风格：

1）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。

2）块的风格：

Java API 源代码选择了行尾风格：



# 第三章 Java基础案例练习：

1）独立编写HelloJava程序，并配上必要的注释。

2）将个人的基本信息（姓名、性别、籍贯、住址）打印到控制台上输出。各条信息分别占一行。

3）结合\n(换行)，\t(制表符)，空格等在控制台打印出如下图所示的效果。

