# 尚马教育 JAVA 基础课程

# Java简介

文档编号：A01

创建日期： 2017-04-12

最后修改日期：2020-12-09

版 本 号：V4.0

电子版文件名：尚马教育-第一阶段-1.Java简介专题课程.docx

**文档修改记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新作者 | 更新说明 | 版本号 |
| 2017-07-30 | 张元林 | 初始版本 | V1.0 |
| 2018-08-01 | 王绍成 | Java基础版本更新 | V2.0 |
| 2019-08-09 | 徐丽莎 | Java基础版本更新 | V3.0 |
| 2020-04-20 | 刘蕾 | Java基础版本更新 | V4.0 |

**主讲人：**

**武汉-刘蕾**

目录

[尚马教育 JAVA 基础课程](#_Toc7221)

[Java简介](#_Toc31565)

[1. 应用程序开发常见概念(了解)](#_Toc27927)

[1.1. 计算机结构与运行原理](#_Toc5557)

[1.2. 程序与编程语言](#_Toc24661)

[1.3. 不同类型语言的编译与运行](#_Toc14267)

[1.4. 进制与编码](#_Toc15789)

[1.5. 字符集](#_Toc12697)

[2. Java语言概述](#_Toc1536)

[2.1. Java语言发展历史](#_Toc29688)

[2.2. Java语言优势](#_Toc13686)

[3. Java开发运行平台](#_Toc10239)

[3.1. JDK(Java开发工具包)](#_Toc24524)

[3.2. JRE(Java运行环境)](#_Toc11480)

[3.3. JVM(Java虚拟机)](#_Toc16453)

[4. Java入门程序](#_Toc20095)

[4.1. 编译到执行的过程分析](#_Toc25028)

[4.2. Java注释](#_Toc26571)

[5. IDE---集成开发环境](#_Toc26911)

[5.1. Eclipse工具](#_Toc29211)

[5.2. 使用步骤](#_Toc6830)

[5.3. 编码规范](#_Toc22285)

## 应用程序开发常见概念(了解)

### 计算机结构与运行原理

* 计算机系统包括硬件和软件两个部分
* 我们通常说的计算机结构指的是硬件部分的结构，了解硬件的结构，有助于我们理解程序的运行过程。

|  |
| --- |
|  |

* 计算机硬件部分包括输入设备、存储器、输出设备、运算器、控制器五大部分
* 计算机工作的简单过程如下:
  + 第一步：将程序和数据通过输入设备送入存储器（可以理解为内存）;
  + 第二步：计算机从存储器中取出程序指令送到控制器（可以理解为CPU）去识别，分析该指令要做什么事;
  + 第三步：控制器根据指令的含义发出相应的命令（如加法、减法），将存储单元中存放的操作数据取出送往运算器进行运算，再把运算结果送回存储器指定的单元中;
  + 第四步：当运算任务完成后，根据指令将结果通过输出设备输出;

### 程序与编程语言

* 什么是程序设计语言？
  + 计算机语言就是为了表达程序而由人设计出来的计算机能接受的人工语言，它是用来表达用户意图、指挥计算机工作的通信工具。
* 什么是程序与指令？
  + 指令：当我们希望电脑为我们做一些事时，必须对它下达命令，这个命令，我们叫做“指令”。

|  |
| --- |
| 例如：   * 新建文件 * 删除文件 * 复制文件 * ….. |

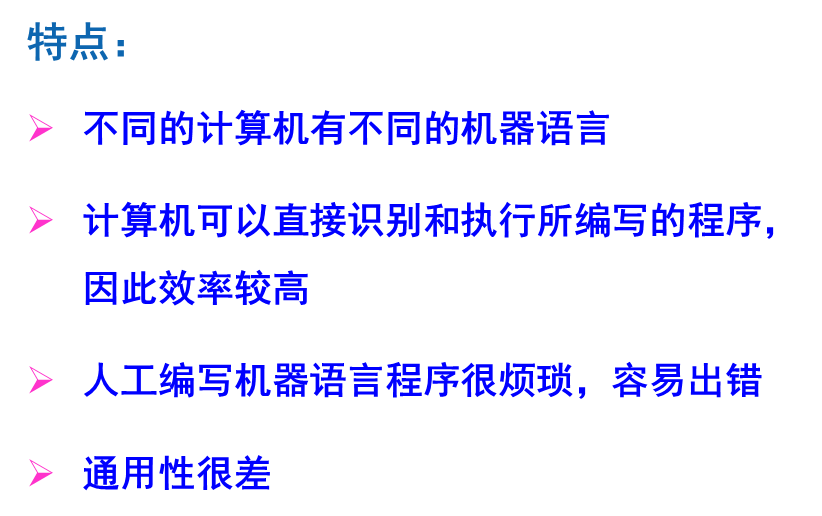
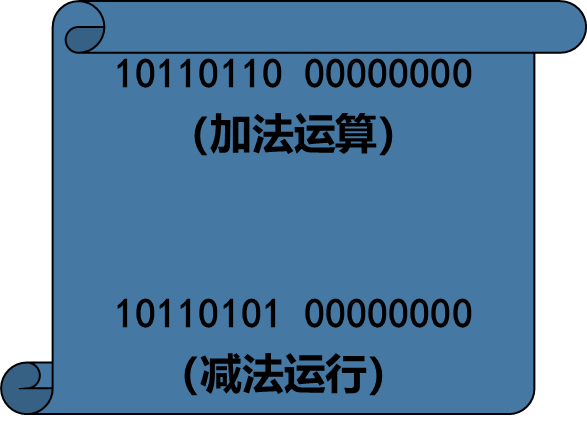
* + 程序：程序是指令的集合，它会告诉计算机执行一系列的任务指令。程序来源于生活，像我们在生活中为达到某目的而做的一系列事情，就是一个程序。

|  |
| --- |
| 例如我们出去旅游，为完成旅游这个程序，我们会执行以下一系列指令。   * 订票 * 持票上车到达目的地 * 观光景点 * …… * 回程 |

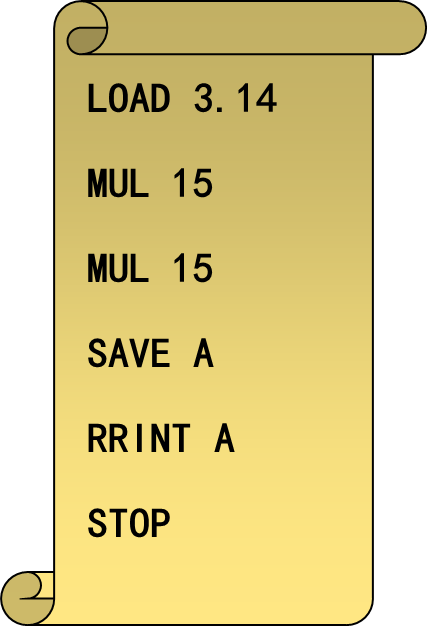
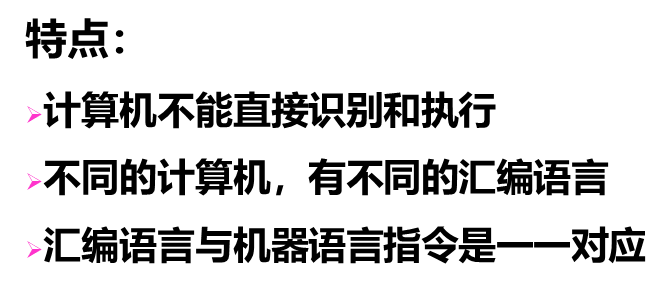
* 我们的学习目标是使用Java语言编写软件；
* 软件包括系统软件（例如 windows操作系统）和应用软件（例如淘宝、京东等）
* 程序是软件的必要元素，任何软件都包含程序，生活中，常常把程序和软件等同，而实际上，程序是软件的一部分
* 程序，也就是使用编程语言进行编写的。有很多种编程语言，Java只是其中一种。
* 编程语言（Programming Language）是定义计算机程序的形式语言
* 编程语言有很多种，例如C、C++、C#、Python、Java……
* 简单的说，编程语言可以看做是程序员和计算机之间交流使用的语言

### 计算机语言的发展历史

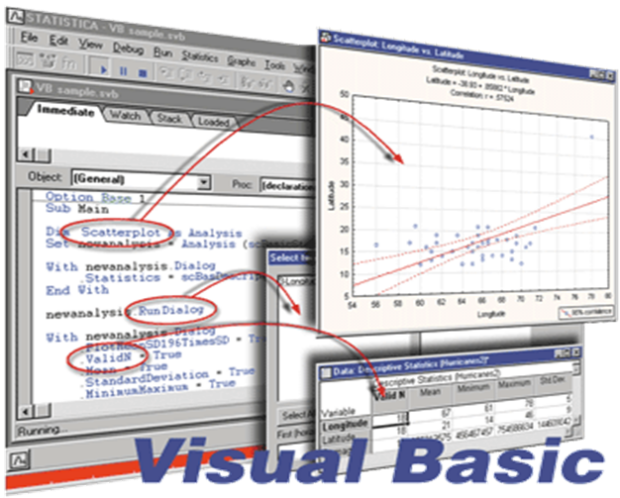
* 计算机语言总的来说分为**机器语言**、**汇编语言**和**高级语言**三大类。
* 第一阶段：机器语言：
  + 机器语言是指一台计算机全部的指令集合。
  + 电子计算机所使用的是由"0"和"1"组成的二进制数，二进制是计算机的语言的基础。计算机发明之初，人们只能降贵纡尊，用计算机的语言去命令计算机干这干那，一句话，就是写出一串串由"0"和"1"组成的指令序列交由计算机执行，这种计算机能够认识的语言，就是机器语言。使用机器语言是十分痛苦的，特别是在程序有错需要修改时，更是如此。
  + 因此程序就是一个个的二进制文件。一条机器语言成为一条指令。指令是不可分割的最小功能单元。而且，由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。机器语言，是第一代计算机语言。



* 第二阶段：汇编语言：
  + 为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如，用"ADD"代表加法，"MOV"代表数据传递等等，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。
  + 汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，所以至今仍是一种常用而强有力的软件开发工具。
  + 汇编语言的实质和机器语言是相同的，都是直接对硬件操作，只不过指令采用了英文缩写的标识符，更容易识别和记忆。它同样需要编程者将每一步具体的操作用命令的形式写出来。
  + 汇编程序的每一句指令只能对应实际操作过程中的一个很细微的动作，例如移动、自增，因此汇编源程序一般比较冗长、复杂、容易出错，而且使用汇编语言编程需要有更多的计算机专业知识，但汇编语言的优点也是显而易见的，用汇编语言所能完成的操作不是一般高级语言所能实现的，而且源程序经汇编生成的可执行文件不仅比较小，而且执行速度很快。

* 第三阶段：高级语言【面向过程和面向对象】
  + 高级语言有：BASIC（True basic、Qbasic、Virtual Basic）、C、C++、PASCAL、FORTRAN、智能化语言（LISP、Prolog、CLIPS、OpenCyc、Fazzy）、动态语言(Python、PHP、Ruby、Lua)等等。- 高级语言源程序可以用解释、编译两种方式执行。通常用后一种。
  + 高级语言是绝大多数编程者的选择。和汇编语言相比，它不但将许多相关的机器指令合成为单条指令并且去掉了与具体操作有关但与完成工作无关的细节，例如使用堆栈、寄存器等，这样就大大简化了程序中的指令。由于省略了很多细节，所以编程者也不需要具备太多的专业知识。　高级语言主要是相对于汇编语言而言，它并不是特指某一种具体的语言，而是包括了很多编程语言，流行的VB、VC、FoxPro、Delphi等，这些语言的语法、命令格式都各不相同。
  + 高级语言的发展：
  + 特别要提到的：在C语言诞生以前，系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；但一般的高级语言又难以实现对计算机硬件的直接操作（这正是汇编语言的优势），于是人们盼望有一种兼有汇编语言和高级语言特性的新语言——C语言。
  + 高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。
  + 高级语言的下一个发展目标是面向应用，也就是说：只需要告诉程序你要干什么，程序就能自动生成算法，自动进行处理，这就是非过程化的程序语言。



### 不同类型语言的编译与运行

* 程序员用编程语言编写的是“源程序”，计算机不能直接运行
* “源程序”要想被运行，必须变成二进制编码才可以
* 不同的编程语言不一样，有的通过“编译”变成二进制编码，有的通过“解释”变成二进制编码，分别被称为“编译型”和“解释型”

|  |
| --- |
|  |

* “编译型”语言是先把源程序的每一条语句都编译成机器语言,并保存成二进制文件, 运行时计算机可以直接以机器语言来运行此程序,速度较快
* “解释型”语言在执行程序时才一条一条的解释成机器语言给计算机来执行,所以运行速度会受到影响

|  |
| --- |
|  |

* Java语言比较特殊，可以说既不是编译型的也不是解释型的；

|  |
| --- |
|  |

### 进制与编码

* 人们平时说的11,24,89…….这样的数，都是逢十进一，用数字0,1-9表示
* X进制指的是：使用0-（X-1）数字表示，逢X进1
* 例如，三进制指的是用0-2表示，逢3进1；0,1,2,10,11（表示十进制的0-4）
* 十进制的0-10，用二进制表示分别为：0,1,10,11,100,101,110,111,1000,1001,1010
* 计算机里的数都使用二进制，也就是说，计算机只认识二进制；
* 二进制表示和阅读起来太长太麻烦，所以编程时也可能使用到八进制、十六进制；
* 八进制用0-7表示，逢8进1；如：0,1,2,3,4,5,6,7,10,11,12,13,14…..
* 十六进制用0-9，A-F表示，逢16进1;如：0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11…..
* 思考:
  + 为什么计算机只能认识二进制？

用二进制原因：

1 . 技术实现简单，计算机是由逻辑电路组成，逻辑电路通常只有两个状态，开关的接通与断开，这两种状态正好可以用“1”和“0”表示。

2. 简化运算规则：两个二进制数和、积运算组合各有三种，运算规则简单，有利于简化计算机内部结构，提高运算速度。

3. 适合逻辑运算：逻辑代数是逻辑运算的理论依据，二进制只有两个数码，正好与逻辑代数中的“真”和“假”相吻合。

4. 易于进行转换，二进制与十进制数易于互相转换。

* + 为什么除了二进制外，比较多用八进制，十六进制？

计算机只认识二进制，8是2的3次方，16是2的4次方，跟二进制的转换容易。

一个二进制，每三位隔一下，用八进制表示，就变为八进制

一个二进制，每四位隔一下，用十六进制表示，就变为十六进制

* 计算机中的符号数有三种表示方法，即原码、反码和补码；
* 三种表示方法均有符号位和数值位两部分，符号位都是用0表示“正”，用1表示“负”，而数值位，三种表示方法各不相同；
* 原码：符号位加上真值的绝对值，即用第一位表示符号，其余位表示值。
* 反码：正数的反码是其本身；负数的反码是在其原码的基础上, 符号位不变，其余各个位取反.
* 补码：正数的补码就是其本身；负数的补码是在其原码的基础上, 符号位不变, 其余各位取反, 最后+1. (即在反码的基础上+1)
* 计算机系统中数都使用补码表示；
* [+1] = [00000001]原 = [00000001]反 = [00000001]补  
  [-1] = [10000001]原 = [11111110]反 = [11111111]补
* 总结
  + 正数的原码、反码、补码相同
  + 负数的原码、反码、补码不同

### 字符集

* 思考:

计算机里不仅仅需要计算处理数字，还需要处理很多其他字符，比如英文，中文，符号等等，是怎么存储的呢？

* 编码： 按照规则将字符存储在计算机中的过程，称为编码。
* 解码： 将存储在计算机中的二进制数解析显示出来，称为解码。【编码的逆过程】

|  |
| --- |
|  |

* 编码与解码之间，必须有一个相对的标准。这个标准一般就称之为：字符集。需要注意的是：并且编码与解码之间的字符集必须一致，否则就会出现乱码。
* 字符集（Charset）：是一个系统支持的所有抽象字符的集合。字符是各种文字和符号的总称，包括各国家文字、标点符号、图形符号、数字等
* 常见的字符集有：
  + ASCII字符集：是基于罗马字母表的一套电脑编码系统。主要是现代英语和其他西欧语言。
  + GB2312字符集：简体中文的字符集。
  + GBK字符集：GB2312的升级版，加入了繁体字。
  + BIG5字符集：是针对繁体汉字的汉字编码。
  + Unicode字符集：国际组织制定的可以容纳世界上所有的文字和符号的字符编码方案。
    - Unicode可以使用的编码有三种，分别为：
    - UTF-8：一种变长的编码方案，使用1-6个字节来存储。
    - UTF-16：固定有可变，占2个字节或这个4个字节来存储。
    - UTF-32：固定不变。不管字符编码大小，始终占用4个字节。--> 提高了效率，浪费了空间。

|  |
| --- |
|  |

* Java语言使用Unicode字符集；Java中的字符串都用Unicode字符集编码。

## Java语言概述

### Java语言发展历史

* “Java” 一词的本意是地名——“爪哇”；爪哇岛位于南太平洋，是印度尼西亚的一部分。
* 全世界三大咖啡产地：巴西 印尼爪哇岛 中国海南岛
* 由此可见，Java的原意是指一个咖啡的产地，也可以说是一种咖啡的品牌  
* Java是一种程序设计语言，由Sun Microsystem公司于1995年推出
* 早在1991年，Sun公司的James Gosling等人开始开发名为Oak的程序设计语言，希望用来控制嵌入在有线电视机顶盒和PDA等设备中的微处理器。代码短小、紧凑且与平台无关，这些要求促使开发团队设计一个可移植的语言，可以为虚拟机生成中间代码。
* 1994年，Oak语言正式更名为Java。
* 1995年5月23日，Java语言诞生。
* 1996年1月，第一个JDK-JDK1.0诞生。
* 1996年4月，10个最主要的操作系统供应商申明将在其产品中嵌入JAVA技术。
* 1996年9月，约8.3万个网页应用了JAVA技术来制作。
* 1997年2月18日，JDK1.1发布。
* 1997年4月2日，JavaOne会议召开，参与者逾一万人，创当时全球同类会议规模之纪录 。
* 1997年9月，JavaDeveloperConnection社区成员超过十万。
* 1998年2月，JDK1.1被下载超过2,000,000次。
* 1998年12月8日，JAVA2企业平台J2EE发布。
* 1999年6月，SUN公司发布Java的三个版本：标准版（JavaSE,以前是J2SE）、企业版（JavaEE以前是J2EE）和微型版（JavaME，以前是J2ME）。
* 2000年5月8日，JDK1.3发布。
* 2000年5月29日，JDK1.4发布。
* 2001年6月5日，NOKIA宣布，到2003年将出售1亿部支持Java的手机。
* 2001年9月24日，J2EE1.3发布。
* 2002年2月26日，J2SE1.4发布，自此Java的计算能力有了大幅提升。
* 2004年9月30日18:00PM，J2SE1.5发布，成为Java语言发展史上的又一里程碑。为了表示该版本的重要性，J2SE1.5更名为Java SE 5.0。
* 2005年6月，JavaOne大会召开，SUN公司公开Java SE 6。此时，Java的各种版本已经更名，以取消其中的数字"2"：J2EE更名为Java EE，J2SE更名为Java SE，J2ME更名为Java ME。
* 2006年12月，SUN公司发布JRE6.0 。
* 2009年04月20日，甲骨文74亿美元收购Sun。取得java的版权。
* 2010年11月，由于甲骨文对于Java社区的不友善，因此Apache扬言将退出JCP。
* 2011年7月28日，甲骨文发布 Java7.0 的正式版。
* 2014年3月18日，Oracle公司发表 Java SE 8。 在近20年中这个版本有了最大的变化。Java8提供了一种“函数式”编程方式，可以很容易地表述并执行的计算。所有的编程语言都必须与时俱进。Java在这方面显示出非凡的能力。

|  |  |  |  |
| --- | --- | --- | --- |
| **Java语言的发展状况** | | | |
| 版本 | 年份 | 语言新特性 | 类与接口的数量 |
| 1.0 | 1996 | 语言本身 | 211 |
| 1.1 | 1997 | 内部类 | 477 |
| 1.2 | 1998 | strictfp修饰符 | 1524 |
| 1.3 | 2000 | 无 | 1840 |
| 1.4 | 2002 | 断言 | 2723 |
| 5.0 | 2004 | 泛型类、“for each”循环、可变元参数、自动装箱、元数据、枚举、静态导入 | 3279 |
| 6 | 2006 | 无 | 3793 |
| 7 | 2011 | 基于字符串的switch、钻石操作符、二进制字面量、异常处理改进 | 4024 |
| 8 | 2014 | lambda表达式、包含默认方法的接口、流和日期/时间库 | 4240 |

* 2017年9月21日，Oracle公司发表 Java SE 9。
* 2018年3月21日，Oracle公司发表 Java SE 10。
* 2018年9月25日，Java SE 11 发布。
* 2019年3月20日，Java SE 12 发布：

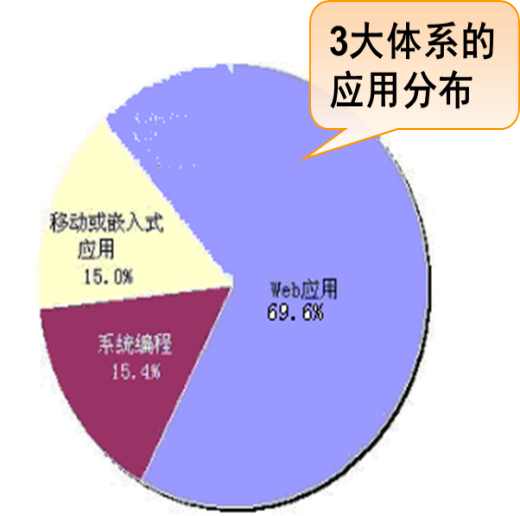
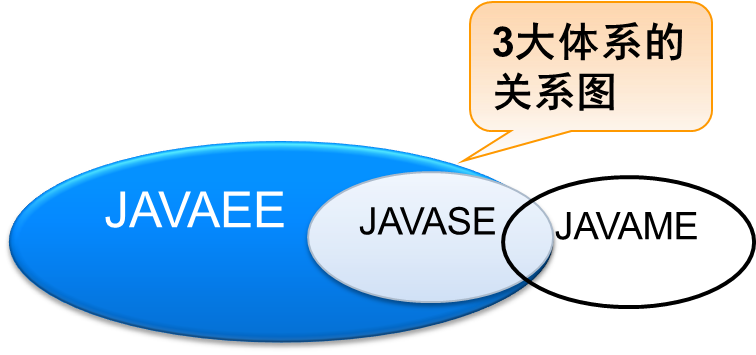
|  |
| --- |
|  |

### Java语言优势

* 跨平台
  + Java是一种既面向对象又可以跨平台的语言，即：编写一次，随处运行；
  + Java不仅适用于单机应用程序和基于网络的程序，也可用于创建消费类设备的附件程序，如移动电话、掌上设备等。
* 简单
  + 去掉了C和C++中许多复杂功能，如指针、运算符重载等，没有goto语句，没有struct和union等
* 面向对象
  + Java是完全面向对象的编程语言，比C++更彻底，纯度更高
* 健壮性
  + 没有指针，避免对指针的误操作造成程序崩溃
  + 程序员分配的内存空间，无需释放，由Java虚拟机的垃圾回收机制来统一完成此项工作，避免了内存泄漏
  + Java的强类型机制保证任何数据必须有明确的数据类型
  + Java提供异常处理机制，能够统一处理异常事件；
* 安全性
  + 由于Java取消了指针运算，有效地提高了程序的安全性
  + Java程序运行在Java虚拟机上，虚拟机可以有效地过滤掉恶意代码，防止程序员有意编写的病毒程序
* 分布性
  + Java程序可以跨平台，跨操作系统，完全支持网络应用程序的设计
  + 提供了用于网络应用编程的类库，包括URL、URLConnection、Socket、 ServerSocket等；
  + Java的RMI(远程方法调用)机制是开发分布式应用的重要手段
* 多线程
  + Java程序使用一个称为“多线程”的进程同时处理多项任务
  + 提供多线程机制允许程序中有多个任务并发执行；
  + 提供的同步机制允许共享数据；

### Java三大特性

* **Java根据应用领域的不同分为三个体系：**
  + **JavaSE（JavaPlatform，StandardEdition）。**JavaSE曾经称为J2SE。它答应开发和布置在桌面、服务器、嵌入式环境和实时环境中使用的Java应用程序。JavaSE包含了支撑JavaWeb服务开发的类，并为JavaPlatform，EnterpriseEdition（JavaEE）供给根底。
  + **JavaEE（JavaPlatform，EnterpriseEdition）。**这个版别曾经称为J2EE。企业版别帮助开发和布置可移植、强健、可伸缩且安全的服务器端Java应用程序。JavaEE是在JavaSE的根底上构建的，它供给Web服务、组件模型、办理和通信API，能够用来完成企业级的面向服务体系结构（service-orientedarchitecture，SOA）和Web2.0应用程序。
  + **JavaME（JavaPlatform，MicroEdition）。**这个版别曾经称为J2ME。JavaME为在移动设备和嵌入式设备（比如手机、PDA、电视机顶盒和打印机）上运行的应用程序供给一个强健且灵敏的环境。JavaME包含灵敏的用户界面、强健的安全模型、许多内置的网络协议以及对能够动态下载的连网和离线应用程序的丰厚支撑。基于JavaME标准的应用程序只需编写一次，就能够用于许多设备，而且能够利用每个设备的本机功能。

## Java的核心机制

### 垃圾回收机制（GC）

* 什么是GC？
  + GC（Garbage Collection)：JAVA/[.NET](https://baike.baidu.com/item/.NET" \t "_blank)中的垃圾回收器。
* Java的垃圾回收机制：
  + Java的内存管理实际上就是对象的管理，其中包括对象的分配和释放。
  + 分配对象使用new关键字；释放对象时，只要将对象所有引用赋值为null即可。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“可用的”,哪些对象是“不可用的”，当GC确定一些对象为“不可用”时，GC就有责任回收这些内存空间。
  + GC在JVM中通常是由一个或一组进程来实现的，它本身也和用户程序一样占用heap空间，运行时也占用CPU，当进程运行时，应用程序停止运行。
  + 因此，当GC运行时间较长时，用户能够感到Java程序的停顿，另一方面，如果GC运行时间太短，则可能对象回收率太低，这意味着还有很多应该回收的对象没有被回收，这即要求我们在设计GC时要均衡效率。
* **总结：有了GC，程序员就不需要手动的去控制内存的释放。但是还是会出现内存溢出的情况。**

### JVM：跨平台

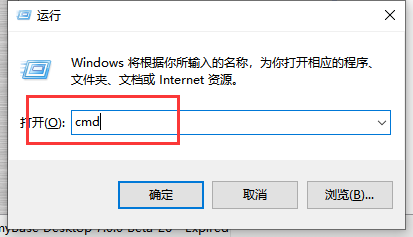
* **什么是跨平台？**
  + 即：编写一次，随处运行！
* **那么Java语言是否跨平台？**
  + Java是一种既面向对象又可以跨平台的编程语言！
* **为什么可以跨平台？**
  + 是因为有JVM（虚拟机）的存在。
* **什么是JVM（虚拟机）？**
  + JVM（Java Virtual Machine）称为Java虚拟机，在Java平台中有着举足轻重的地位；
  + JVM可以理解为Java编译器和操作系统间的虚拟处理器；
  + 编译器编译出的字节码只要JVM认识即可；
  + JVM再将字节码解释成操作系统认识的机器码；
  + 使用Java语言编写的程序，实际上是运行在JVM之上，而不是运行在操作系统上；
  + 它有一个解释器组件，可以实现Java字节码和计算机操作系统之间的通信。

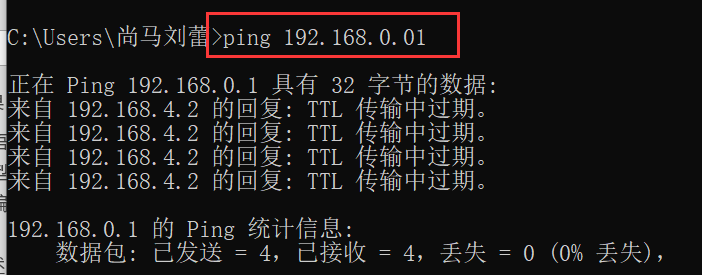
|  |
| --- |
|  |
| 因为有了JVM，所以同一个Java程序在三个不同的操作系统中都可以执行。这样就实现了Java程序的跨平台性。也称为Java具有良好的可移植性。  Java语言是跨平台的，而JVM不是跨平台的。【每个系统平台都有自己的虚拟机】 |

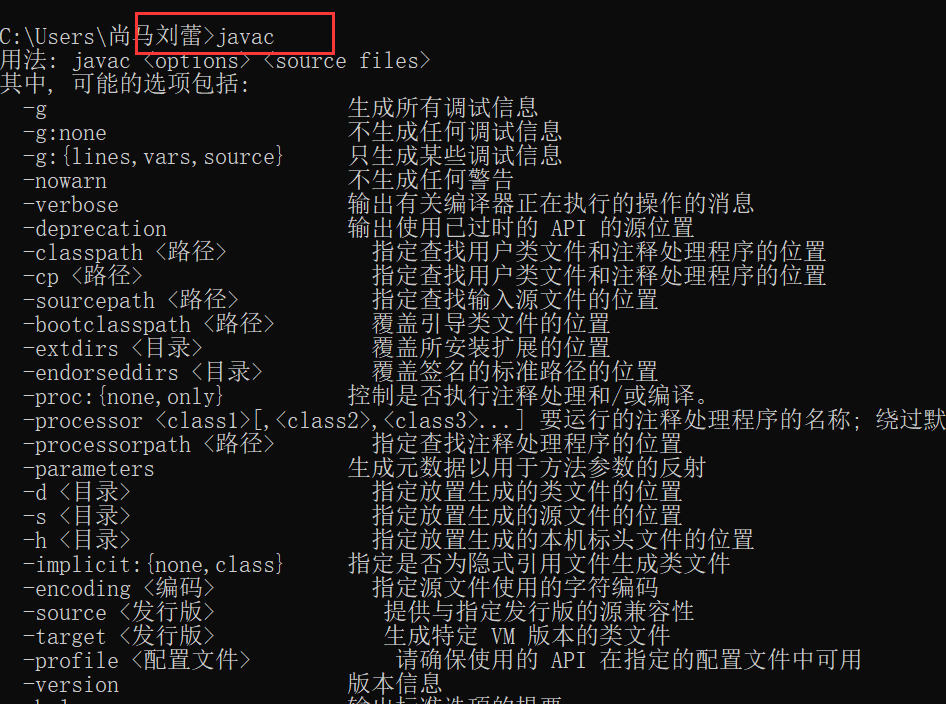
## Java入门程序

### 常用DOS命令：

* **什么是DOS命令？**
  + 计算机术语，是指DOS操作系统命令，是一种面向磁盘的操作命令。
* **启动DOS命令窗口：** 
  + 通过window+R键，启动dos窗口。输入cmd命令，进入dos命令窗口。



* **常用的DOS命令有：**
  + PING命令：专用于TCP/IP协议的探测工具。 
  + D:  --> 进入对应的盘符。
  + dir：列出当前目录下所有的文件及文件夹。
  + md：创建目录/文件夹。
  + rd：删除目录/文件夹。【只能删除空的文件夹】
  + cd：进入目录。
  + cd..：退回上一级目录。
  + cd\：退回到根目录。
  + exit：退出dos命令窗口。
  + cls：清屏。
  + 。。。
* **Java中很重要的DOS命令：**
  + **java或者javac**：检查环境变量是否配置成功了？



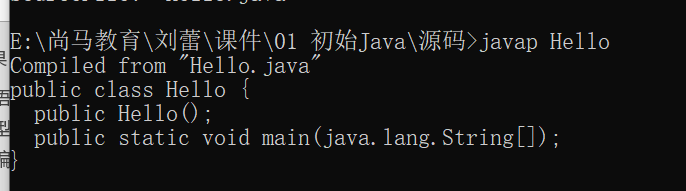
* + 编译源文件：**javac 源文件名称【需要后缀名】**



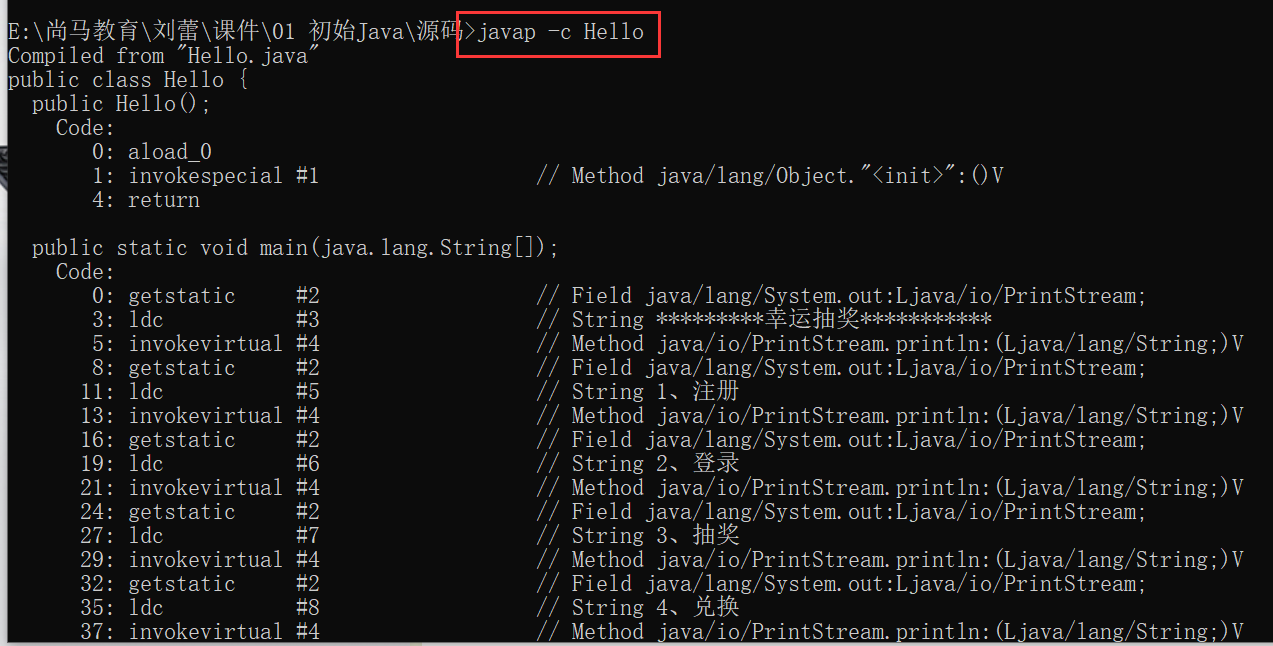
* + 运行字节码文件：**java 字节码文件名称【没有class后缀名】**



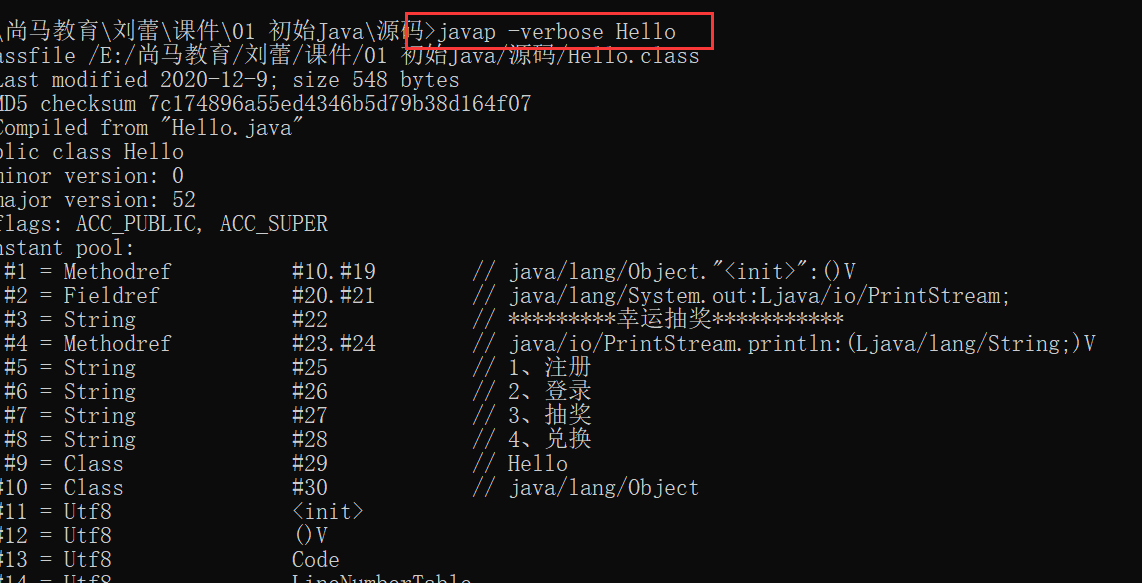
* + 反编译：Javap 🡪 主要用于帮助开发者了解Java编译器的机制。
    - 查看Java编译器生成的字节码：javap 字节码文件名称【没有class后缀名】



* + - 分解方法代码，即显示每个方法具体的字节码：javap -c 字节码文件名称【没有class后缀名】



* + - 指定显示更进一步的详细信息。

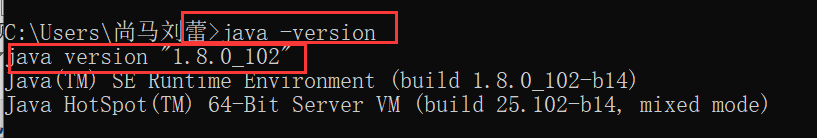


* + 生成文档：javadoc java文件名称【需要后缀名】

****

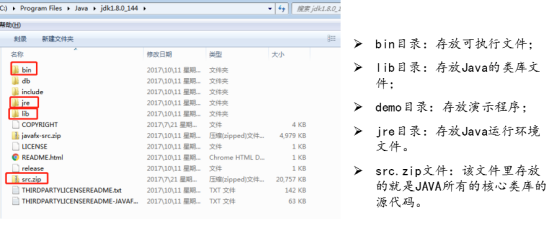
### 安装和卸载JDK

* **下载JDK**： <https://www.oracle.com/cn/java/technologies/javase-downloads.html>
* **安装JDK**： 基本上就是下一步、下一步。
* **测试是否安装成功？** 
  + 方式一：在安装目录下查看。
  + 方式二：使用dos命令的方式查看。 java -version【查看jdk对应的版本】



* **卸载JDK：和普通软件卸载一样。**

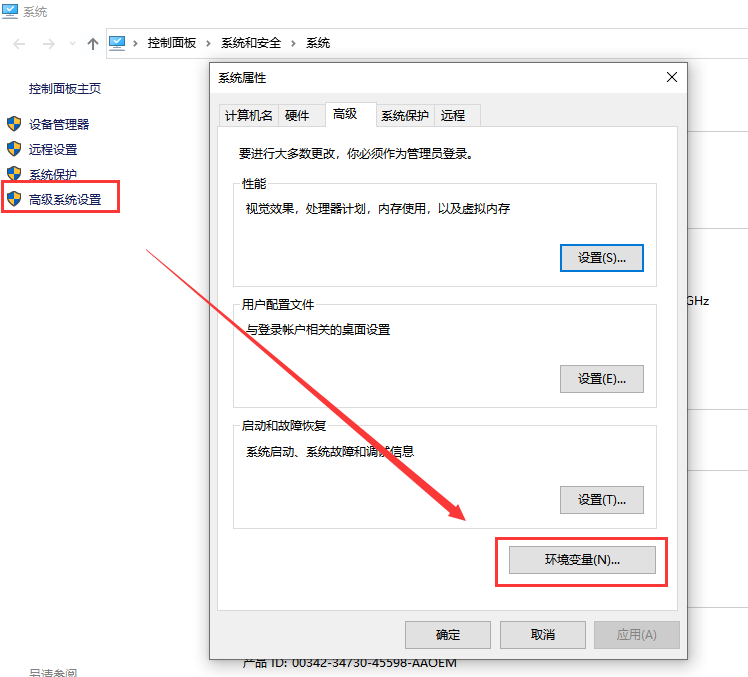
### JDK和JRE

* **什么是JDK?** 
  + JDK：Java Development kit  --> Java开发工具包。
  + 简单的理解为开发者需要使用的。比如：编译（javac命令）、运行（java运行命令）等都是存放在JDK中的。     
* **什么是JRE？** 
  + JRE：Java Runtime   Environment --> Java运行环境。
  + 简单的理解为使用者所需要的。   --> 由于程序员既需要编写又需要运行，所以需要安装Sun公司提供的开发工具包。【自带安装Jre】

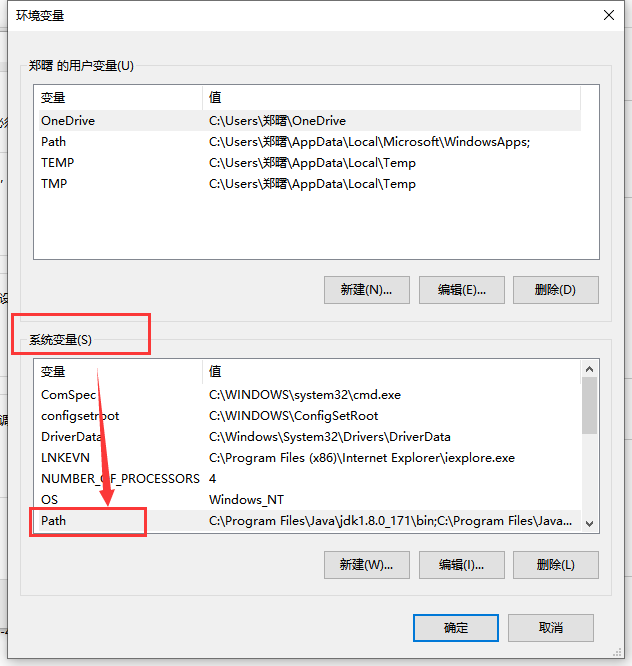
### 配置环境变量

{作为开发人员来说，必须掌握。}

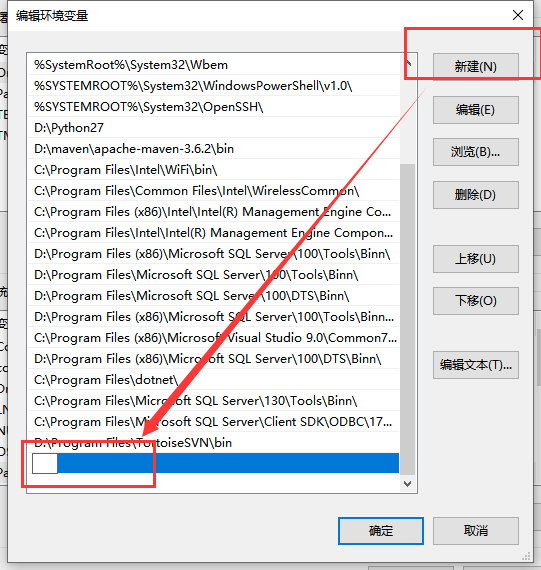
* **为什么需要配置环境变量？** 
  + 环境变量：使某个文件夹的应用程序在任意路径下都可以执行。
  + 简单的理解为就是希望JDK这个路径：C:\Program Files\Java\jdk1.8.0\_171\bin【JDK安装的路径】下的所有命令在DOC命令中可以执行。
* **如何配置环境变量？** 
  + 我的电脑 --> 右键属性  --> 高级系统设置



* + 系统变量：Path



* + 双击Path进入，新建。然后将此路径：C:\Program Files\Java\jdk1.8.0\_171\bin复制到空白处即可。【最好上移到最顶部，方便查看】



* + 检测环境变量是否配置成功？
    - 在dos窗口中，输入命令：javac 【参考4.3中的dos命令】

### 编写第一段代码

* 三步走:

|  |
| --- |
|  |

* + 创建Java源程序
    - Java源程序可以使用任何文本编辑器创建与编辑，一般用.java作为扩展名，其实就是一个使用Java语言编写的文本文件；
  + 编译源程序
    - Java编译器读取Java源程序并翻译成Java虚拟机能够明白的指令集合，并以字节码的形式保存在文件中，通常，字节码文件以.class作为扩展名；
  + 运行class（字节码）文件
    - Java解释器读取字节码文件，取出指令并翻译成计算机能够执行的机器代码，完成运行过程。
* 具体流程:
  + 打开记事本，新建一个文本文件，将文件名改为HelloWorld.java，在文件中键入如下内容并保存

|  |
| --- |
|  |
| * 关键字解析：   + public：公共的。🡪 访问修饰符，用来控制类成员的可见范围及作用域。   + class：类。 🡪 用来定义一个类。   + HelloWorld：类名，自定义。🡪 必须与Java源文件中的名称一致。   + static：静态的 🡪 允许在不创建类的实例的情况下，调用main函数。   + void：无返回值。   + main：main入口。 🡪 程序的入口，一个程序只能有一个入口。   + String []：字符串数组，S必须大写，中阔号必须是英文的符号。   + args：字符串数组名称，自定义名称。   + System.out.print()：输出语句，输出之后不换行。   + System.out.println()：输出语句，输出之后换行。 |
|  |

* + 顺利通过编译后将生成一个扩展名为.class的字节码文件（类描述文件）。
    - 打开cmd窗口，运行javac命令，编译源文件
    - 格式: javac 源文件名称

|  |
| --- |
|  |
|  |

* + 运行程序
    - 在成功编译之后，调用java命令，执行程序。
    - 格式: java 字节码文件的文件名称

|  |
| --- |
|  |

### 编译到执行的过程分析

* 以HelloWorld.java为例，编译到执行的过程如下所示：

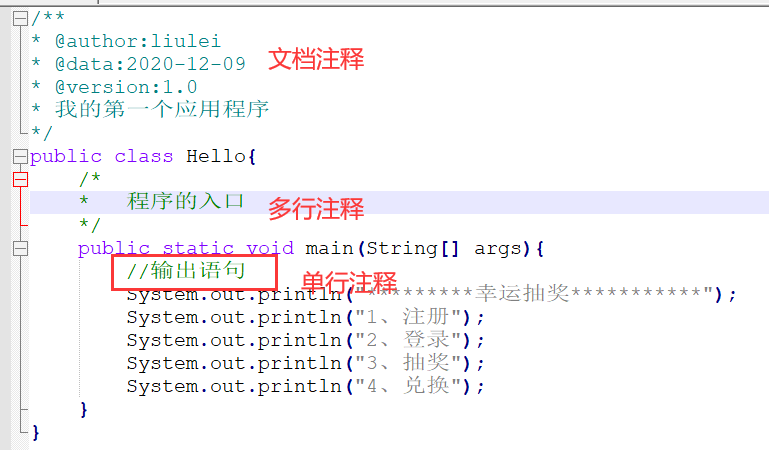
|  |
| --- |
|  |

* 需注意：
  + {}是成对出现的，整个类及其所有成员都包含在一对花括号中（即{和}之间），它们标志着类定义块的开始和结束。
  + System.out.println(); 一行代码结束之后要用分号;隔开，再写下一行。
  + 在Java中所有的单词都是区分大小写的，关键字必须严格按照要求。所有发符号都是英文的符号，除输出符号以外。
  + 在Java中，类名称一般首字母大写。

### Java注释

与其它编程语言一样，也可以在Java文件添加注释，Java文件中的注释有三种：

* 单行注释：以//开始，在行尾结束 🡪 只能注释一行内容。说白了就是给当前的这一行添加注释。
* 多行注释：以/\*开始，以\*/结束，可以有多行。🡪 包含在/\* 和 \*/之间，能注释很多行的内容。提高了注释的可读性。
* 文档注释：以/\*\*开始，以\*/结束，可以有多行。🡪 包含在/\*\* 和 \*/之间，也能注释多行内容。一般用在类、方法的上面，用来描述其作用等等信息。并且注释后，鼠标方法方法名或者类上，可以显示对应的注释内容。



## IDE---集成开发环境

* 在实际开发工作中，不会使用记事本来开发，虽然记事本确实可以用来写代码；
* 都会使用一种集成开发环境（ IDE ）进行开发，IDE会提供很多辅助功能，提高开发效率
* 常用的能用于Java开发的IDE有
  + Eclipse 官方网站：https://eclipse.org/ 基于插件开发，对JavaEE支持较好；
  + NetBeans 官方网站https://netbeans.org/基于插件开发，对JavaSE支持较好；
  + Idea 官方网站：https://www.jetbrains.com/idea/features/ 谷歌将其作为Android开发首选IDE
  + MyEclipse 官方网站:http://www.genuitec.com/products/myeclipse/
  + 本课程使用Eclipse。

### Eclipse工具

|  |
| --- |
|  |

### 使用步骤

* 步骤1：创建工程 file-new-java project

|  |
| --- |
|  |

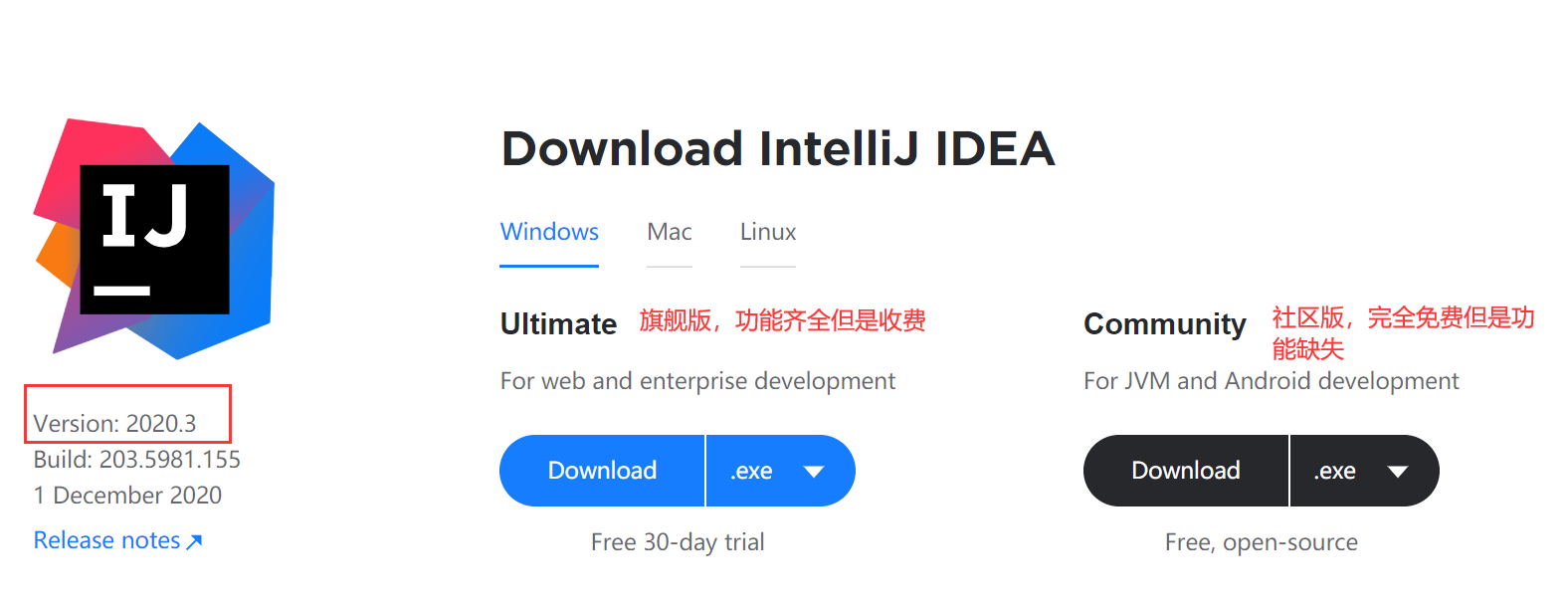
* 步骤2：创建Java类 右键点击src，new-class，创建源程序

|  |
| --- |
|  |

* 步骤3：完成Java类，右键点击HelloWorld.java，选择 run as-Java Application/F11

### IDEA安装与破解

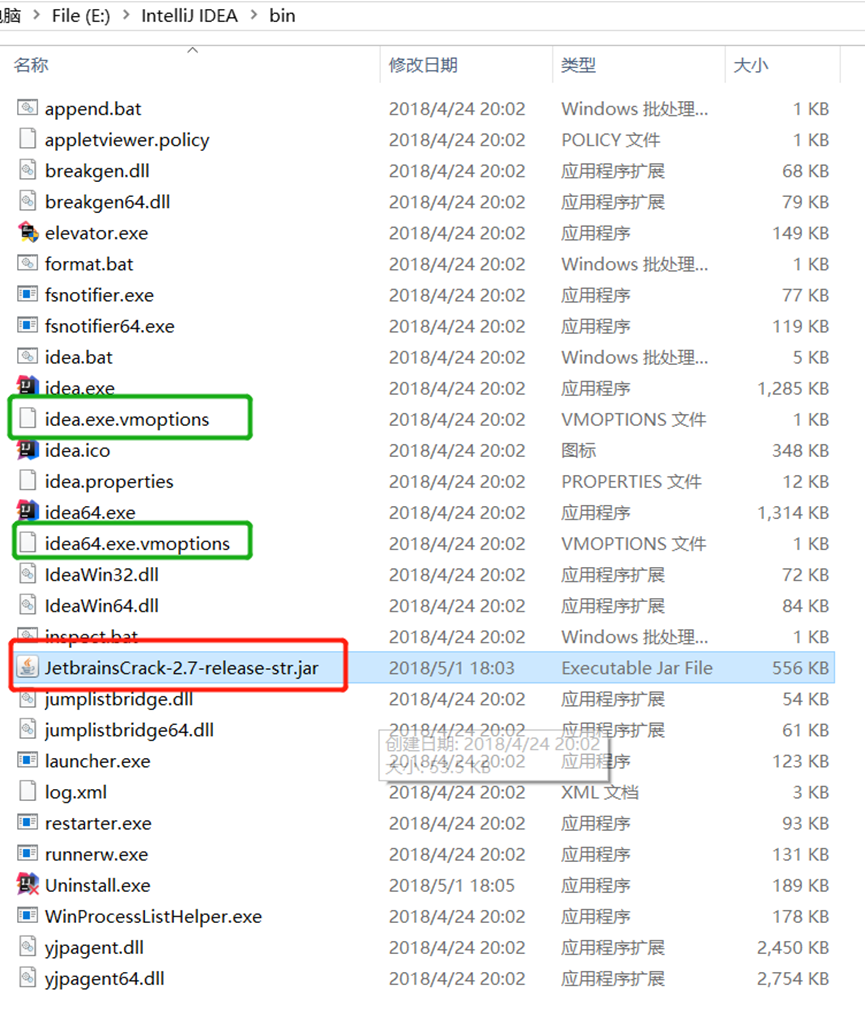
* **下载地址：**https://www.jetbrains.com/idea/download/#section=windows



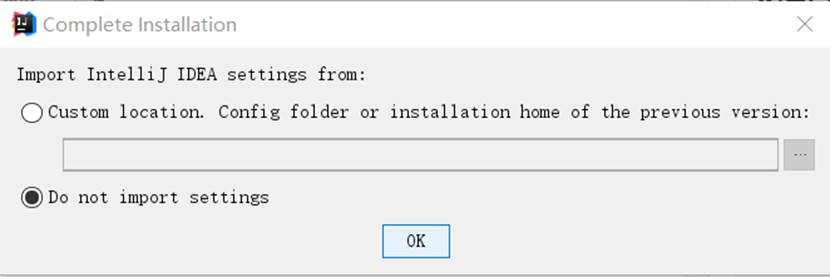
* **资金允许的情况下，建议购买正版。** https://www.jetbrains.com/idea/buy/#commercial?billing=yearly
* **使用注册码，但是过段时间就会过期。不怕麻烦的过期了再注册一次就可以了。**

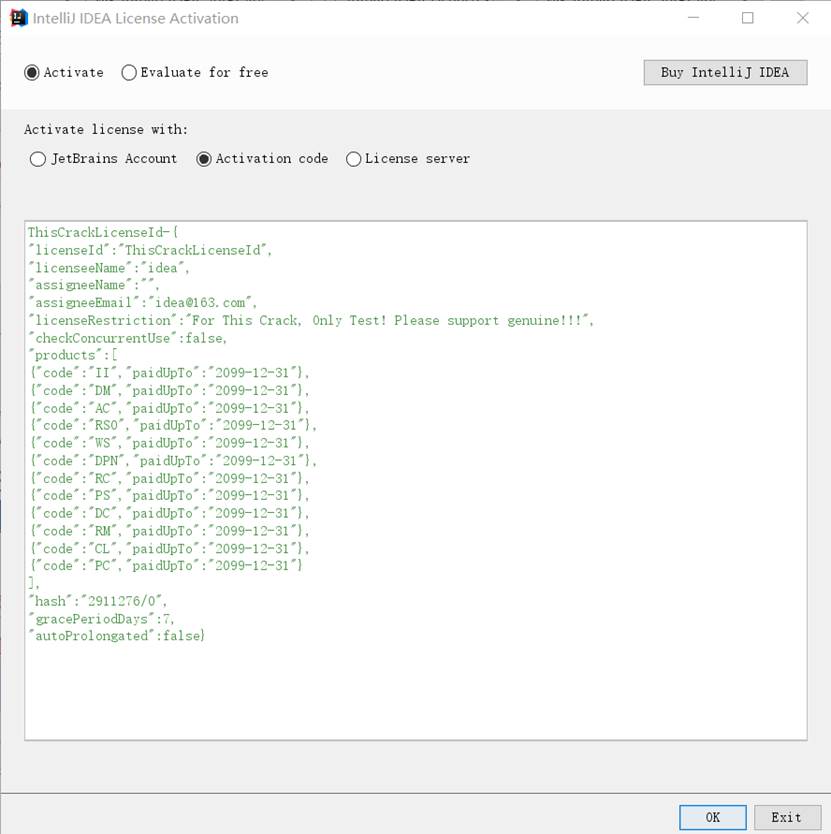
http://idea.lanyus.com/

* **最好的方式：破解旗舰版。**
  + 将JetbrainsCrack-2.7-release-str.jar拷贝到你的Idea安装目录的bin目录下。



* + - 在安装的idea下面的bin目录下面有2个文件 ： 一个是idea64.exe.vmoptions，还有一个是idea.exe.vmoptions（如上图绿框）。
    - 用记事本打开，分别在最下面一行增加一行：-javaagent:E:\IntelliJ IDEA\bin\JetbrainsCrack-2.7-release-str.jar
    - 注意：“-javaagent:E:\IntelliJ IDEA\bin\JetbrainsCrack-2.7-release-str.jar”是我对应的JetbrainsCrack-2.7-release-str.jar的位置,你用你自己的安装路径。
  + 启动软件：
    - 如果是第一次安装，如图选择



* 第一个选择是购买了正版的，第二个选择是我们要用的，第三个选择是修改服务器的                  

下面是需要复制的代码

ThisCrackLicenseId-{

"licenseId":"ThisCrackLicenseId",

"licenseeName":"idea",

"assigneeName":"",

"assigneeEmail":"idea@163.com",

"licenseRestriction":"For This Crack, Only Test! Please support genuine!!!",

"checkConcurrentUse":false,

"products":[

{"code":"II","paidUpTo":"2099-12-31"},

{"code":"DM","paidUpTo":"2099-12-31"},

{"code":"AC","paidUpTo":"2099-12-31"},

{"code":"RS0","paidUpTo":"2099-12-31"},

{"code":"WS","paidUpTo":"2099-12-31"},

{"code":"DPN","paidUpTo":"2099-12-31"},

{"code":"RC","paidUpTo":"2099-12-31"},

{"code":"PS","paidUpTo":"2099-12-31"},

{"code":"DC","paidUpTo":"2099-12-31"},

{"code":"RM","paidUpTo":"2099-12-31"},

{"code":"CL","paidUpTo":"2099-12-31"},

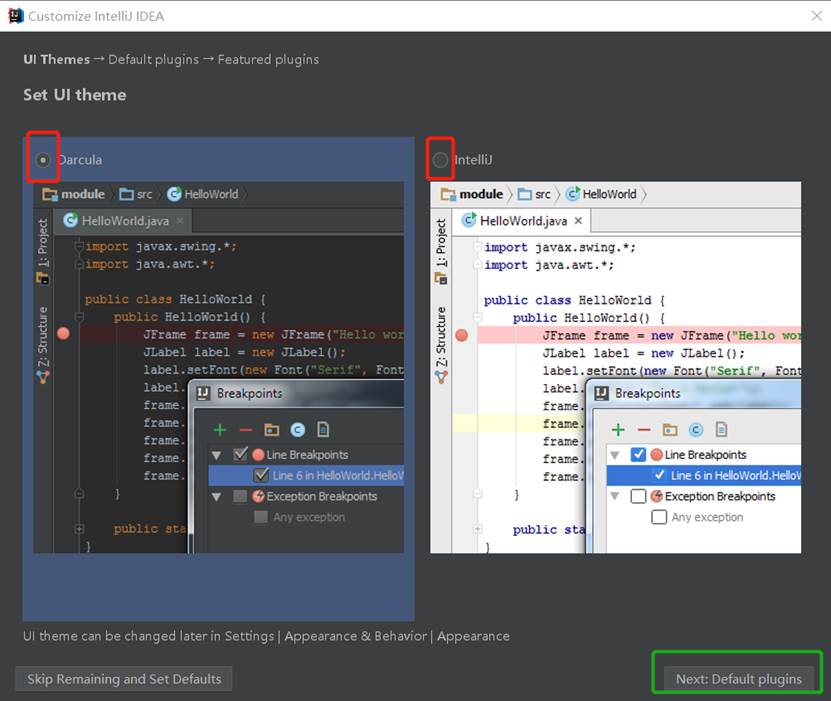
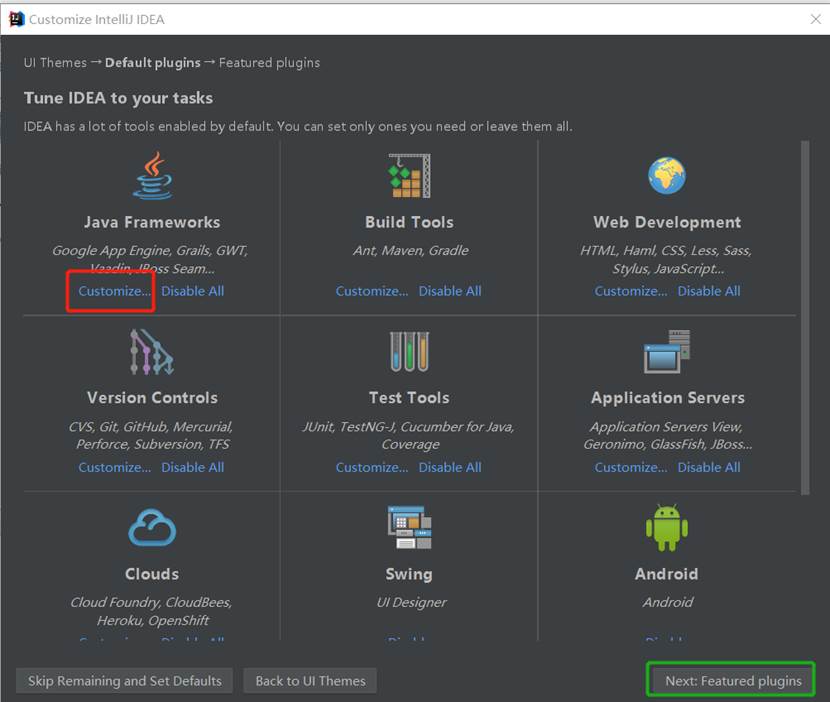
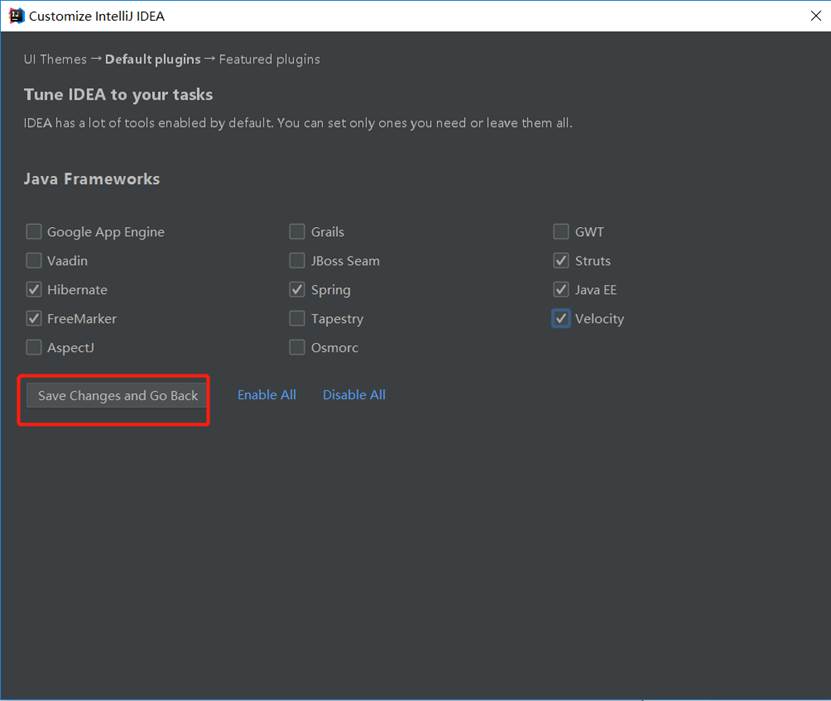
{"code":"PC","paidUpTo":"2099-12-31"}

],

"hash":"2911276/0",

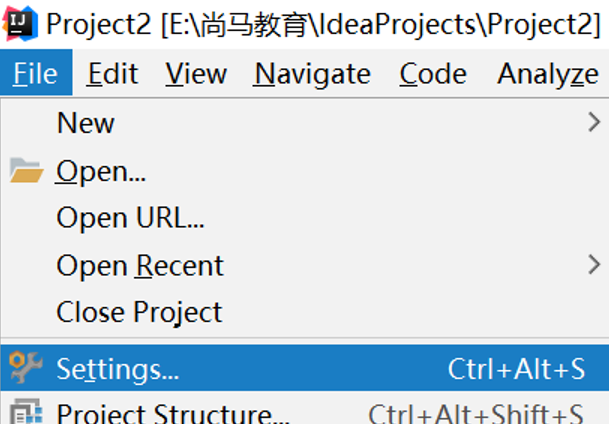
"gracePeriodDays":7,

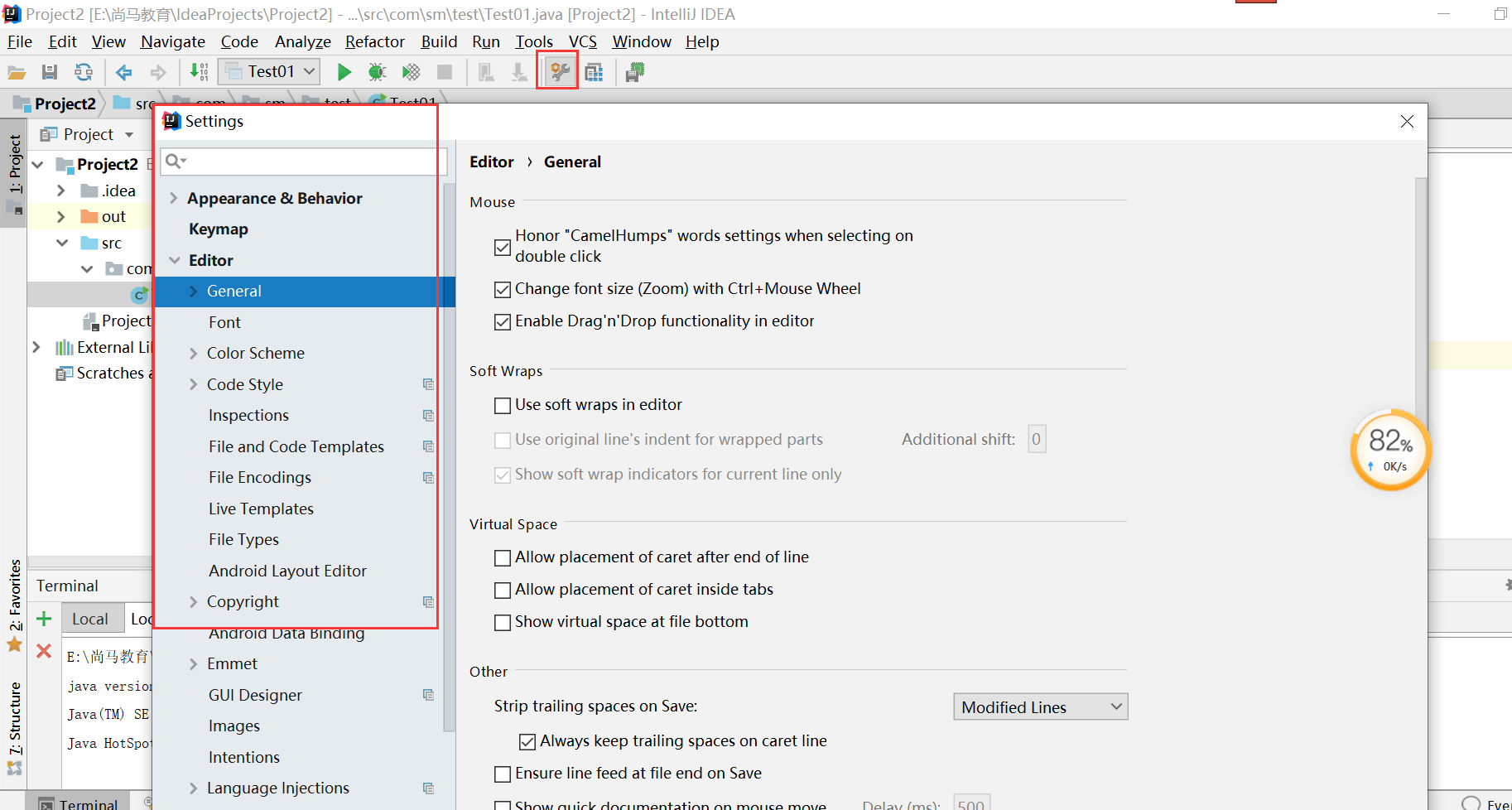
"autoProlongated":false}

* 如果破解成功，可以正常启动。下图红色框是选择模板，根据个人喜好，绿色框是下一步                      
* 下图是设置IDE支持的哪些功能和插件，根据需求选择，这里我们选择java。
* 下图是选择默认的插件，根据个人需要 

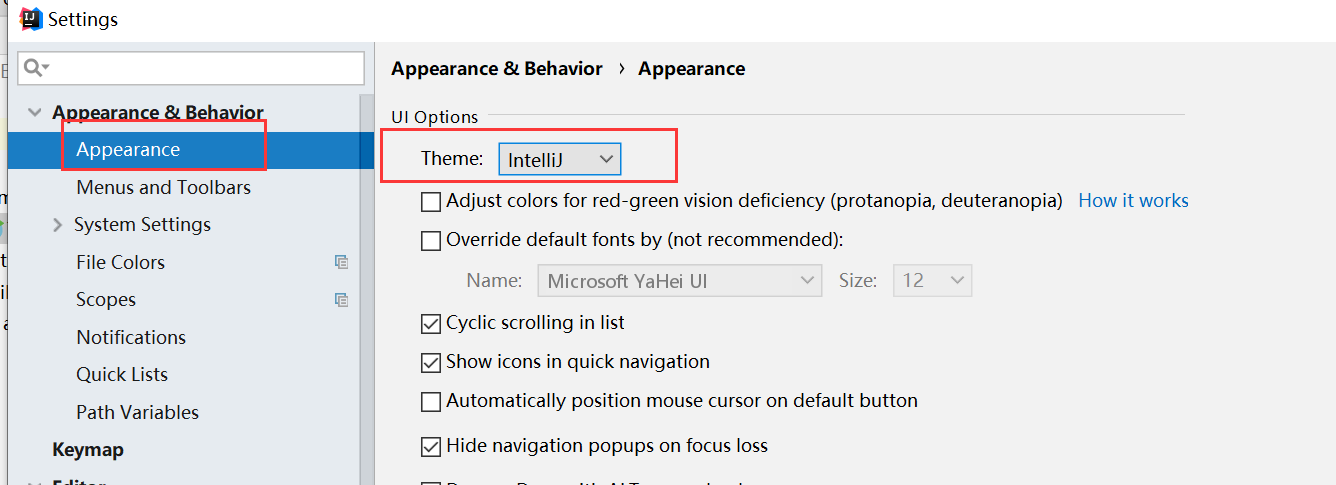
### IDEA常用设置

* 进入设置：

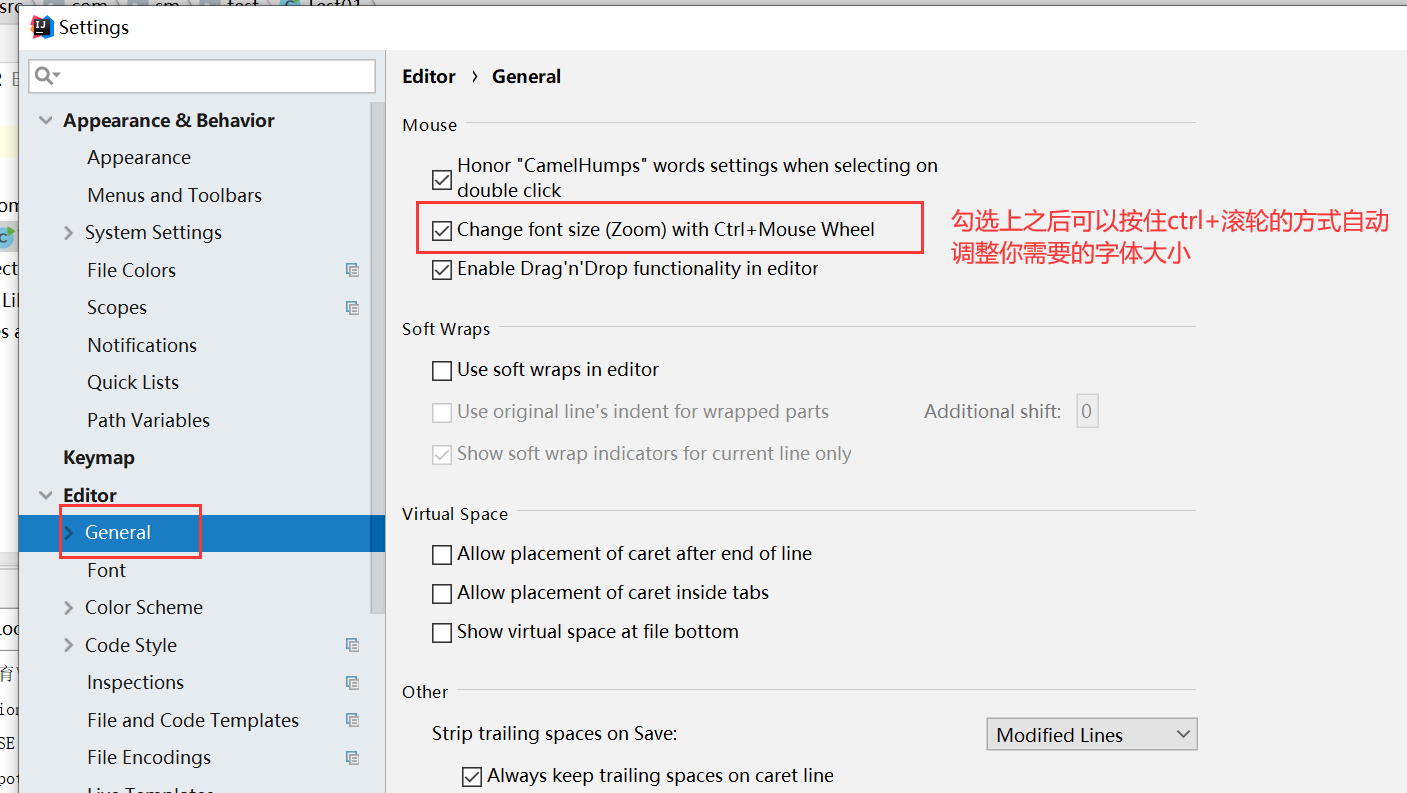




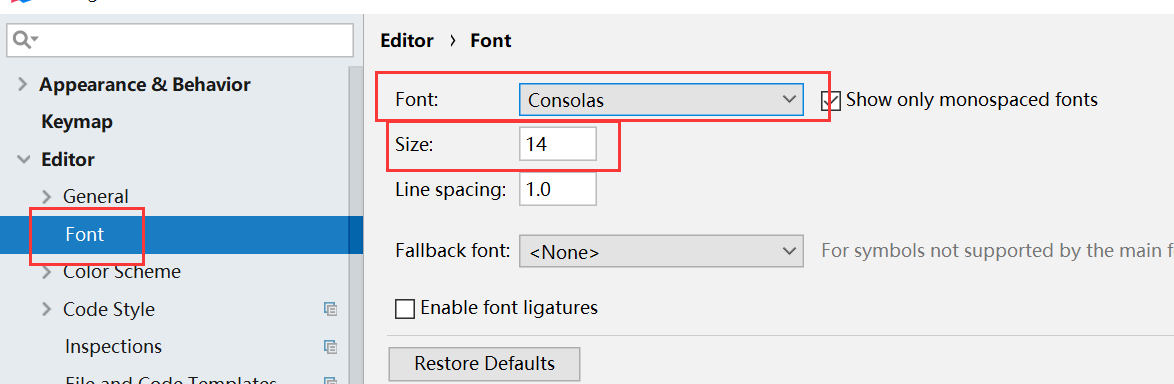
* 设置主题。



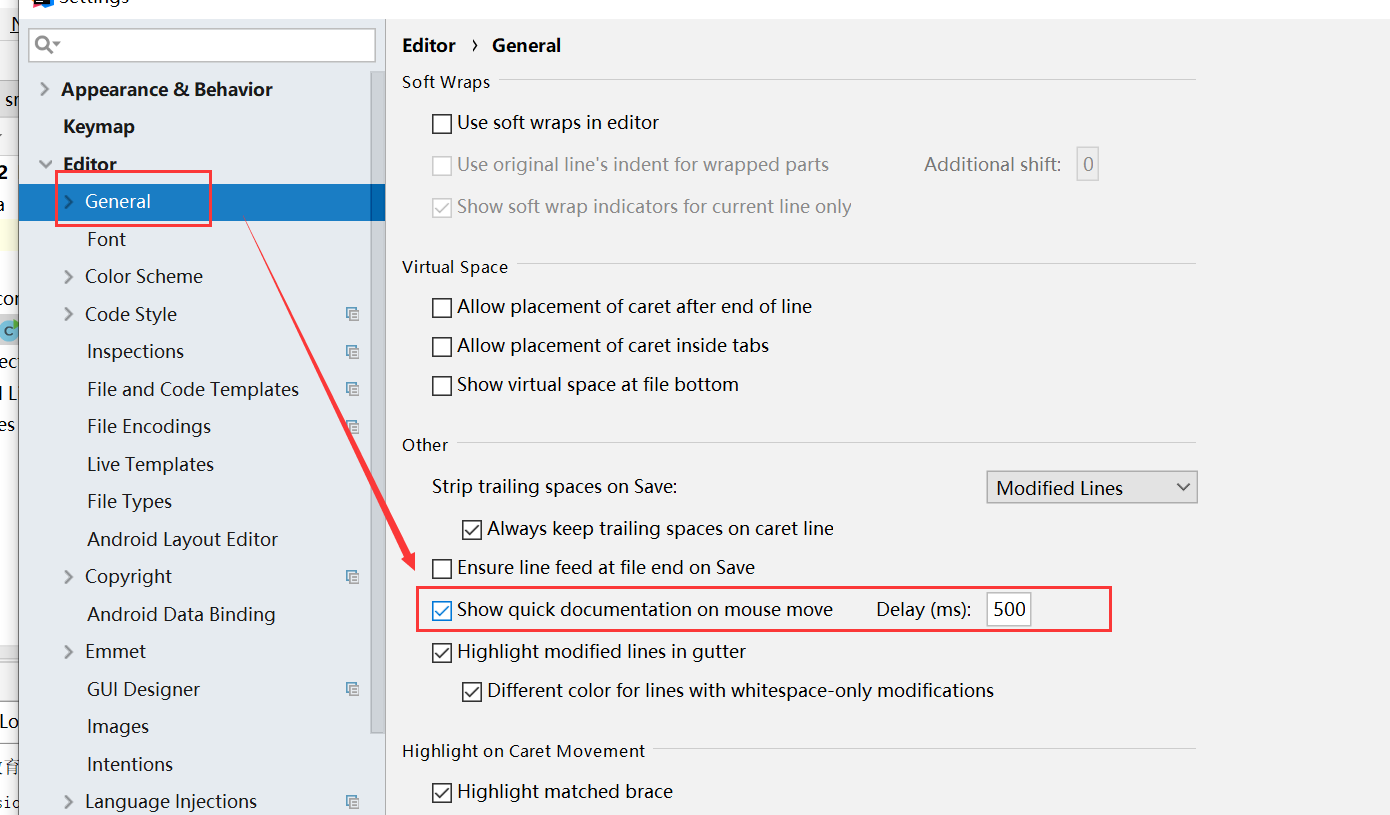
* 更改字体大小。



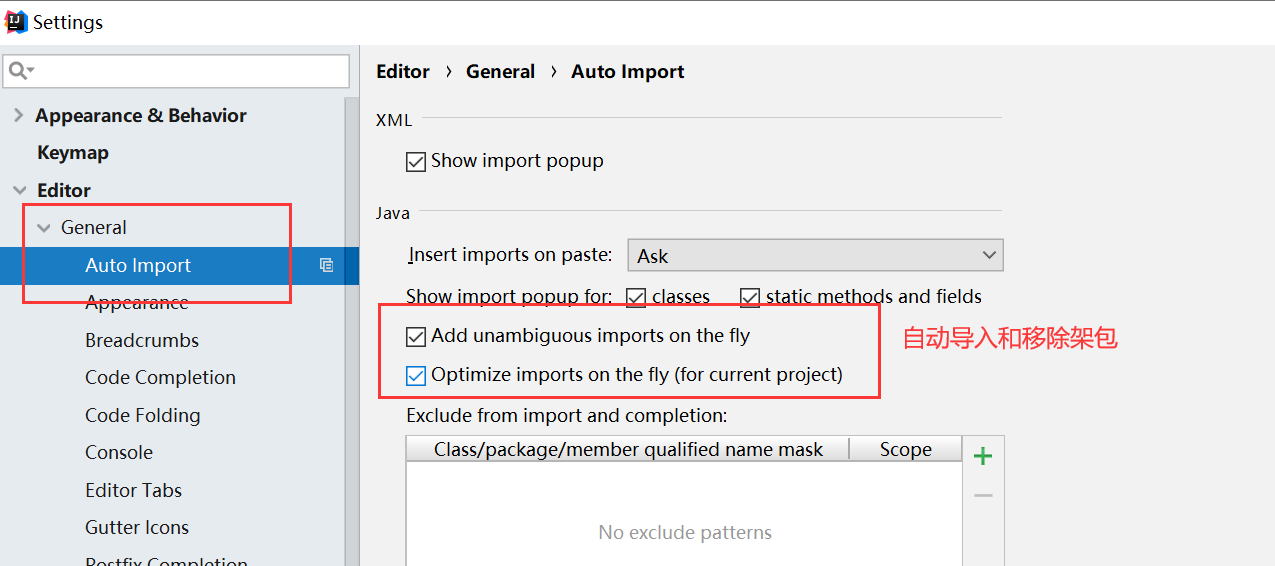
* 或【进入默认的字体和字体大小】



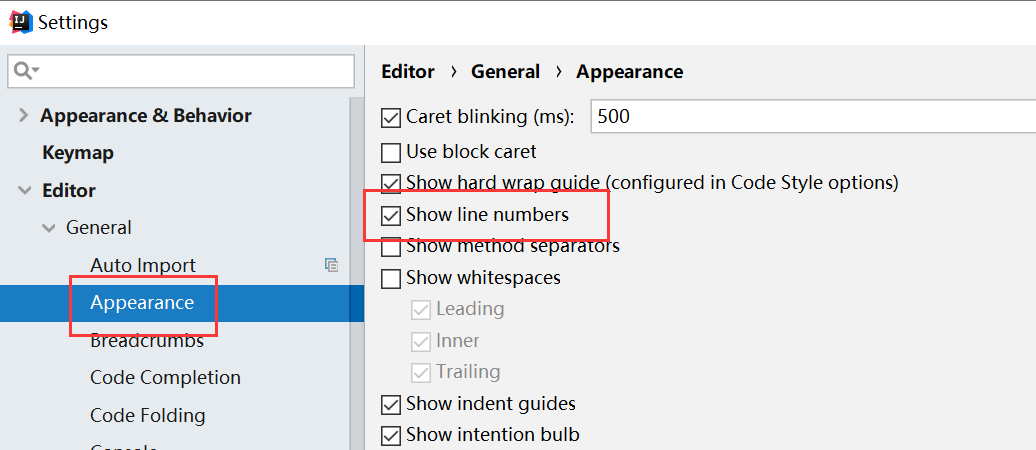
* 设置鼠标悬停，显示代码对应的提示信息。【500为多少时间之后显示对应的提醒，单位为毫秒】



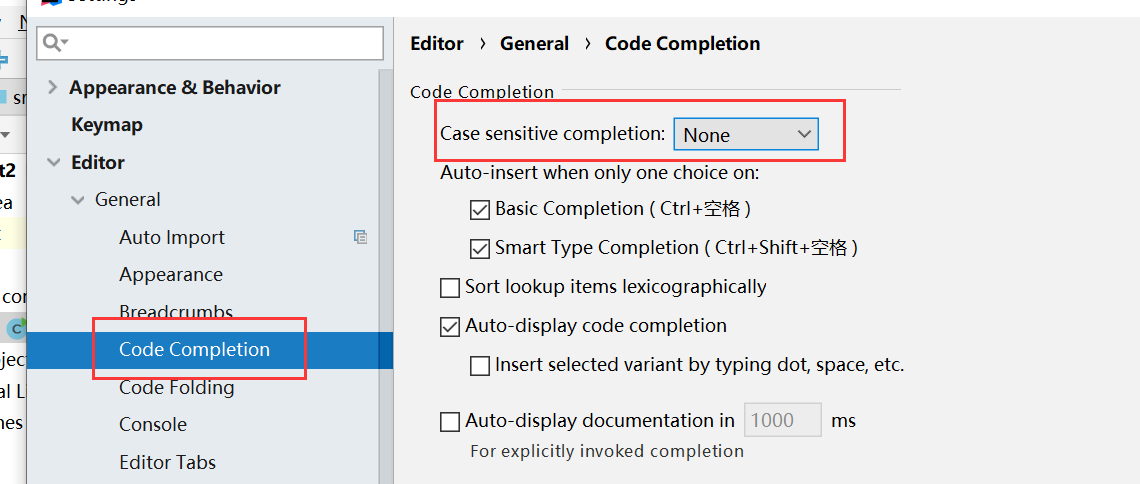
* 自动导入和移除不同的架包：



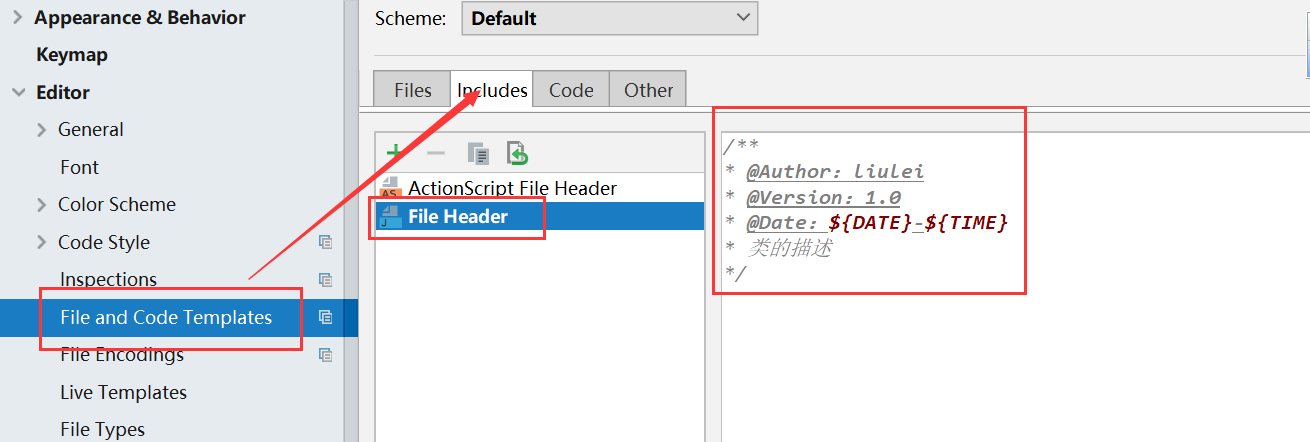
* 显示和隐藏行号：



* 忽略大小写/大小写不敏感：



* 给新建的类添加默认的注释。

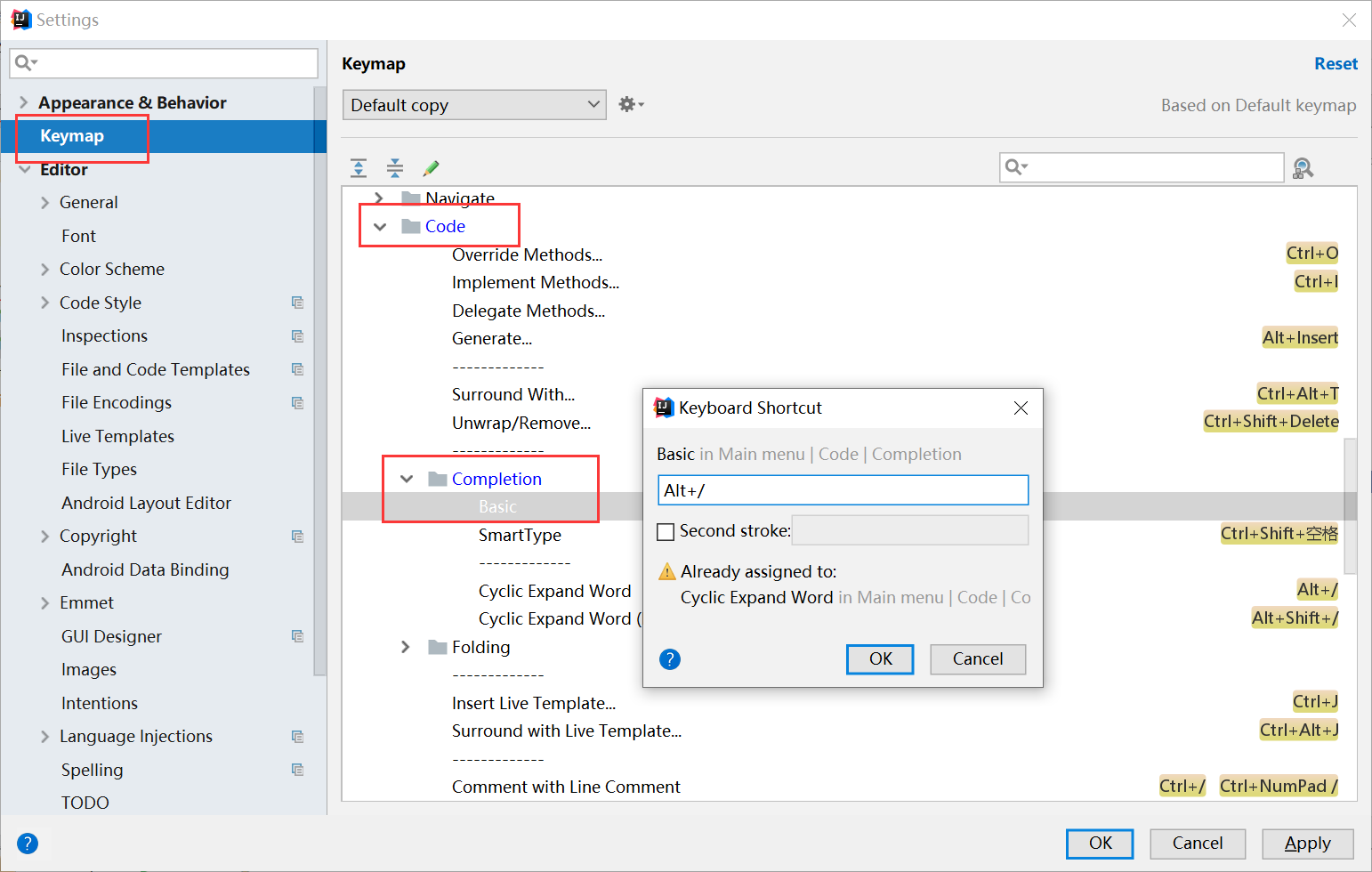


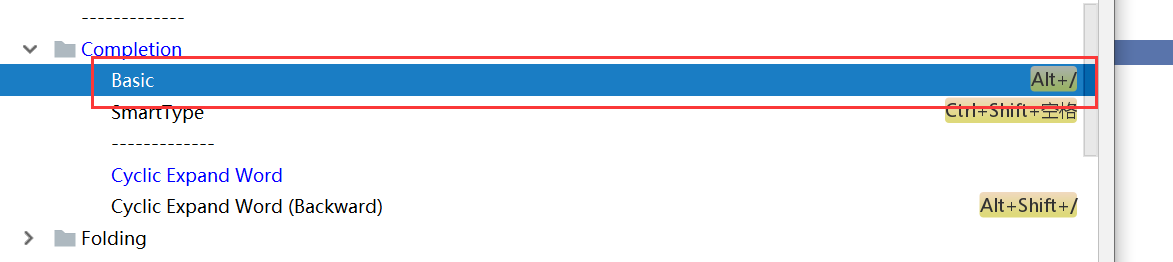
* /\*\*
* \*@Author：liulei
* \*@Version：1.0
* \*@Date：${DATE}-${TIME}
* \*类的描述
* \*/

\*/

### IDEA常用快捷键

* 快速创建main方法：psvm
* 快速输出语句：sout  或者 变量.sout
* 复制一行：ctrl+D
* 删除一行：ctrl+y
* 撤回：ctrl+z
* 代码上下移动：ctrl+shift+上下键
* 代码左右选中：ctrl+shift+左右键
* 快速选中某个关键字/某个单词：双击即可。
* 自动生成代码：alt+insert。 --> 比如类的get和set方法等。
* 万能的快捷键：alt+enter --> 手动导入架包{没有设置自动导入架包的时候才有效果}、自动生成变量、返回值等。
* 快速生成注释：ctrl+/  【单行注释】 ctrl+shift+/【多行注释】 --》 取消注释和快速注释快捷键是一样的
* 添加一个代码自动补全的快捷键，常用的是alt+/





* 快速排版/代码格式化：ctrl+alt+L  --> 和QQ的快捷键冲突了，取消QQ的就行了。

### 编码规范

* 类名命名方式: 驼峰式命名 首字母大写第二个单词首字母大写
* 包名命名方式: 公司域名后缀名+公司名称+其他
* [www.javasm.cn](http://www.javasm.cn) cn/com/edu/org.javasm.其他
* public修饰的类的名称必须与Java文件同名!
* main方法作为程序入口， void必不可少！必须是public static void main(String[] args)
* Java对大小写敏感，严格区分大小写。
* 每一条Java语句必须以英文分号结束!