Day02 - 开发环境和第一个Java程序

今日学习内容

- 计算机语言
- 跨平台原理
- 开发和运行环境的搭建
- HelloWorld程序开发
- IDEA开发工具
- Java的基本语法
- 数据类型、常量

今日学习目标

- 了解什么是计算机语言
- 了解什么是java的跨平台原理
- 掌握如何搭建开发和运行环境
- 必须掌握HelloWorld程序的开发
- 熟悉使用IDEA工具来开发
- 必须记住Java的基本语法
- 了解注释符号有哪些
- 了解什么是关键字和保留字
- 必须记住java的分隔符和标志符有哪些
- 掌握数据类型和常量

第二章 Java基础入门

1、Java和跨平台

1.1 计算机语言和编程相关概述（了解）

1.1.1 计算机语言（了解）

在生活中的两个人需要交流，无非是采用一种彼此都能够识别的语言。那么，我们说该语言是他们传递信息的媒介。那么什么是计算机语言呢？计算机语言是指用于人与计算机之间通讯的一种特殊语言，是人与计算机之间传递信息的媒介。

为什么需要和计算机交流呢？计算机怎么能读懂我们给它发出的信息？和计算机交流的目的，就是让计算机帮我们完成复杂工作（**重复性、程序化**），比如大量数据的运算。为了让计算机能读懂我们发出的信息，此时就需要编写一套由字符、数字所组成并按照某种语法格式的一串串计算机指令，而这些指令和命令就是计算机语言。

所以，通过计算机语言，人类可以命令计算机帮我们完成复杂的工作。

1.1.2 什么是编程（了解）

我们已经知道，计算机语言就是用来实现人和计算机通讯的，那为什么人要和计算机通讯呢，其原因就是为了让计算机帮我们完成一些人做起来比较复杂的工作。

那计算机怎么知道我们要它解决的问题是什么，怎么知道解决问题的具体的步骤是什么呢？此时我们就得通过计算机语言去告诉计算机：**需要做什么**，**怎么一步一步去做**。这种人和计算机之间交流的过程，我们称之为**编程**。

思考题：我们应该如何学编程？

```
1> 明确需求
2> 分步骤用计算机语言告诉计算机如何一步一步完成需求
```

1.1.3 Java语言（了解）

1990年末，Sun公司预料嵌入式系统将在未来家用电器领域大显身手。于是Sun公司成立了一个由James Gosling领导的"Green计划"，准备为下一代智能家电（电视机、电话等）编写一个通用控制系统。但对于硬件资源极其匮乏的单片式系统来说，C/C++程序过于复杂和庞大,另外由于消费电子产品所采用的嵌入式处理器芯片的种类繁杂，如何让编写的程序跨平台运行也是个难题。

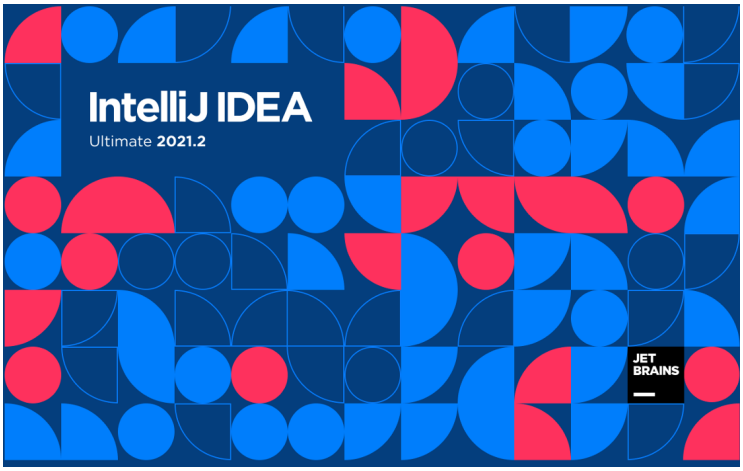
为此，团队成员决定创造一种全新的语言(Oak)来解决**程序跨平台运行的难题**，当需要处理器芯片支持时，硬件生产商并未对此产生极大的热情。因为他们认为，在所有人对Oak语言还一无所知的情况下，就生产硬件产品的风险实在太大了。

1995年，互联网的蓬勃发展给了Oak机会。业界急需一种软件技术来开发一种程序，这种程序可以通过网络传播并且能够跨平台运行，James Gosling意识到这个机遇后，团队就开始正式维护开发这门语言了。

Java诞生于1995年，原属于SUN公司，2009年4月20日，美国数据软件巨头甲骨文公司（**Oracle**）宣布以74亿美元收购SUN公司。Java是最受欢迎的后台开发语言，已经火了30+年，并将继续引领着IT行业的编程语言。Java的LOGO是一杯热气腾腾的咖啡，真的是令人回味无穷。

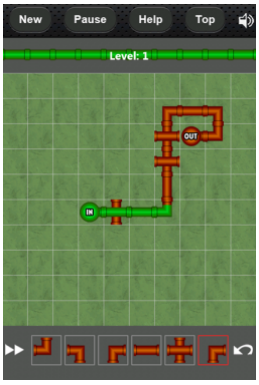
Java的三大平台：Java SE、Java EE，Java ME （**知道即可**）

- Java SE：Java标准平台，它允许开发软件运行在电脑桌面上(用户需要下载，安装)，我们未来使用的IDEA开发工具就属于桌面端应用。 JavaSE适合做桌面端应用 ，同时JavaSE也是JavaEE的基础。



- Java EE：Java企业平台，它允许开发的软件运行在服务器上，针对**Web方向**，主要应用于开发企业项目和互联网项目，如淘宝，京东，12306，各大银行网站等。JavaEE 适合做服务器端开发

- Java ME （了解）：Java微型平台，用来开发早期嵌入式移动设备上的软件，比如早期功能机上的Java游戏。



思考题：

小米/华为等Android系统中的程序使用Java语言的哪个版本开发的?

你会发现，这个世界除了空气无处不在之外，还有Java。

Feb 2020	Feb 2019	Change	Programming Language	Ratings	Change
1	1		Java	17.358%	+1.48%
2	2		C	16.766%	+4.34%
3	3		Python	9.345%	+1.77%
4	4		C++	6.164%	-1.28%
5	7	^	C#	5.927%	+3.08%
6	5	v	Visual Basic .NET	5.862%	-1.23%
7	6	v	JavaScript	2.060%	-0.79%
8	8		PHP	2.018%	-0.25%
9	9		SQL	1.526%	-0.37%
10	20	^^	Swift	1.460%	+0.54%
11	18	^^	Go	1.131%	+0.17%
12	11	v	Assembly language	1.111%	-0.27%
13	15	^	R	1.005%	-0.04%
14	23	^^	D	0.917%	+0.28%
15	16	^	Ruby	0.844%	-0.19%
16	12	vv	MATLAB	0.794%	-0.40%
17	21	^^	PL/SQL	0.764%	-0.05%
18	14	vv	Delphi/Object Pascal	0.748%	-0.32%
19	13	vv	Perl	0.697%	-0.40%
20	10	vv	Objective-C	0.688%	-0.76%

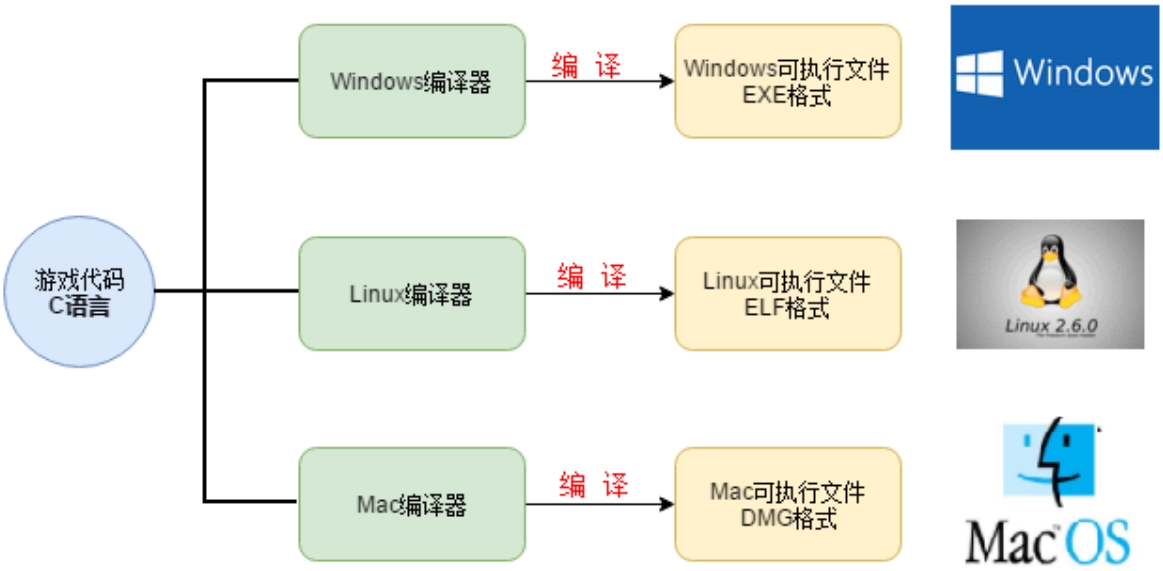
1.2 跨平台原理

1.2.1 平台相关性（了解）

Windows系统只能运行Windows的程序（exe），Linux系统只能运行Linux的程序，Mac系统只能运行Mac的程序。

Windows的可执行文件，不能直接运行在Linux系统中，反之亦然，就好比Android手机不能运行iPhone的App程序一样，我们把这种情况称之为平台相关性。

1.2.2 跨平台性（掌握）

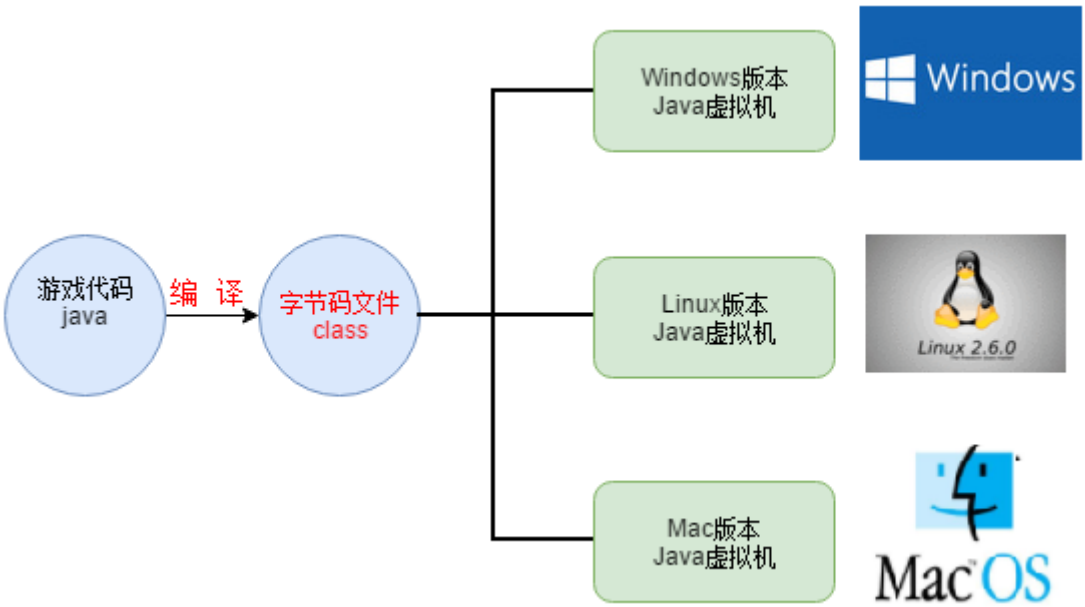


一份（源）代码，多次编译，到处运行

使用特定平台的编译器所编译的程序只能在对应的平台运行，此时会发现编译器是与平台相关的，编译后的文件也是与平台相关的。

我们说的**语言跨平台是编译后的文件跨平台。**

那语言的跨平台性如何实现？拿Java举例，我们可以对Java程序进行编译操作，编译后生成一种和平台系统无关的文件——字节码文件。但是此时Windows、Linux是不能执字节码文件的，只有Java虚拟机（JVM）才能识别字节码文件，那么为了在Windows系统上运行该Java程序，我们就只能在Windows平台上安装Windows版本的JVM，如果要在Mac系统上运行，那么得安装Mac版本的JVM。

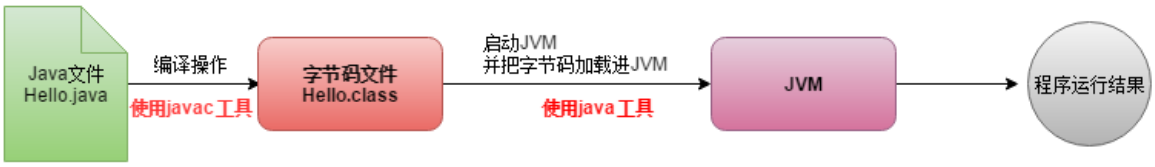


那么如此一来，Java就实现了跨平台，也就达到了**“一次编译，到处运行”**的效果。

Java之所以能跨平台有两个原因：

- Java文件经过编译后生成和平台无关的class文件
- Java虚拟机（JVM）是不跨平台的

在这里进行编译操作的工具叫做javac，启动JVM并把字节码加载进JVM的工具叫做java。

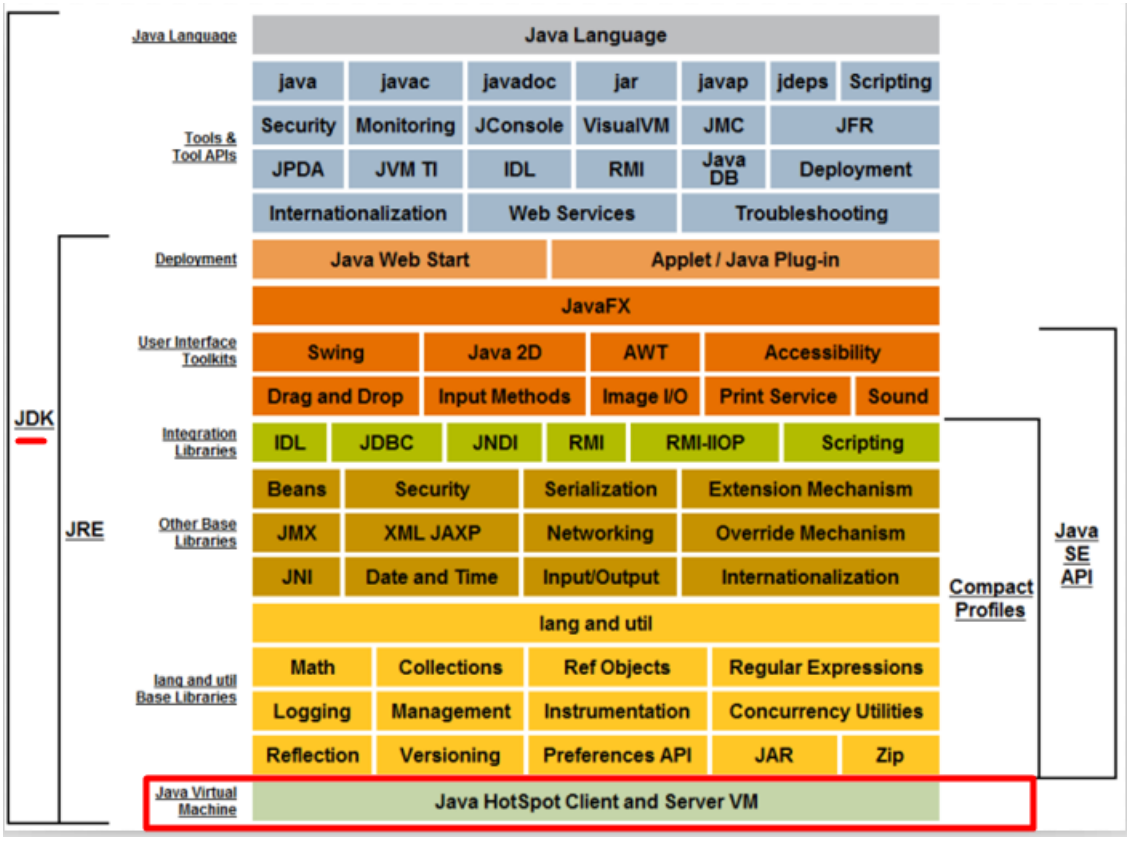


这里解释几个术语：

- 源代码：简称源码，是指还未编译的按照编程语言规范书写的代码，比如Java代码
- 源文件：存放源代码的文件，比如Java文件，拓展名是.java
- 字节码文件： 经过编译器预处理过的一种文件,是JAVA的执行文件存在形式，拓展名是.class

2、Java开发和运行环境搭建（重点）

2.1 JVM、JRE和JDK概述（了解）



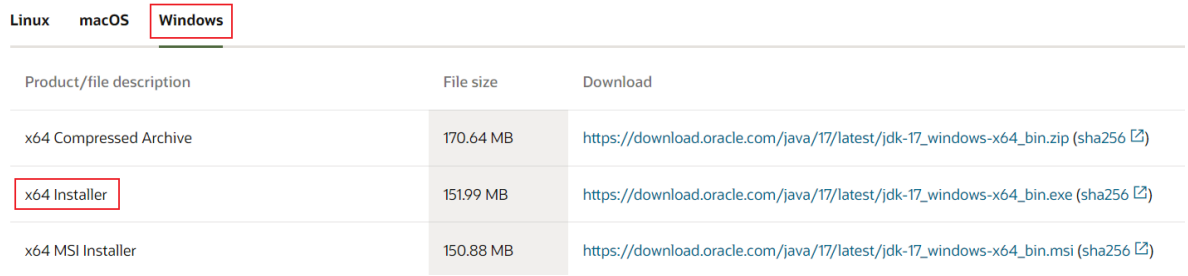
- JVM（Java Virtual Machine）:Java虚拟机，它是运行所有Java程序(Java字节码)的虚拟计算机。JVM是不跨平台的，在Windows下装Windows版的JVM，在Linux下装Linux版的JVM。
- JRE(Java Runtime Environment)：Java运行环境，如果要运行Java程序，就需要JRE的支持。JRE里包含JVM，还包含运行程序需要的常用类库，一般在只运行程序而不开发程序的服务器中安装。
- JDK(Java Development Kit)：Java开发工具包，包含开发Java程序的所有工具如javac和java等。JDK包含JRE，如果已经安装了JDK就不必安装JRE。

```
jvm:java虚拟机，专门用于执行java的字节码文件。
jre:java程序的运行环境，包含jvm + 常用类库
jdk:java程序的开发环境， 包含jre + javac/java等编译运行工具。
```

2.2 安装JDK（掌握）

当出现以下界面的时候，说明当前系统不具备Java的开发环境，就得去安装JDK和配置环境变量。

选择和操作系统一致版本的JDK，如window 64位选择【 Windows x64 Installer 】,mac选择【 macOS Installer】



选择适合自己电脑的JDK后，全称傻瓜式的下一步，其中需要注意的是：

- JDK不要安装在C盘，且不要带中文的路径，路径中不包含空格字符
- 安装完JDK后，JRE不用安装

举例说明：

正例：D:\OpenSources\jdk1.8.0_45\bin

反例：D:\Java开发\jdk1.8.0_45\bin　　出现中文

反例：D:\Program Files\jdk1.8.0_45\bin　　路径中出现空格

注意：在JDK安装目录下的bin目录中，存在编译工具（javac）和运行工具（java）。

2.3 配置PATH环境变量（掌握）

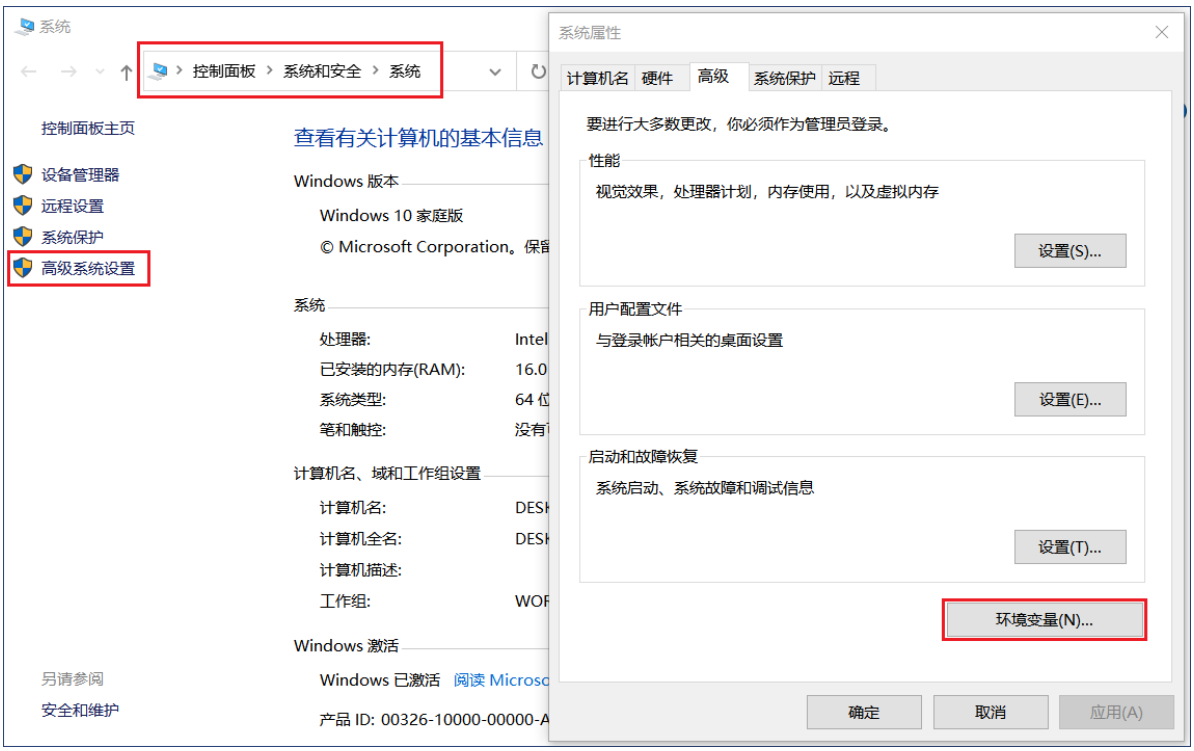
PATH环境变量的作用

配置PATH环境变量的目的是，能够在任何地方使用编译工具（javac）和运行工具（java）。

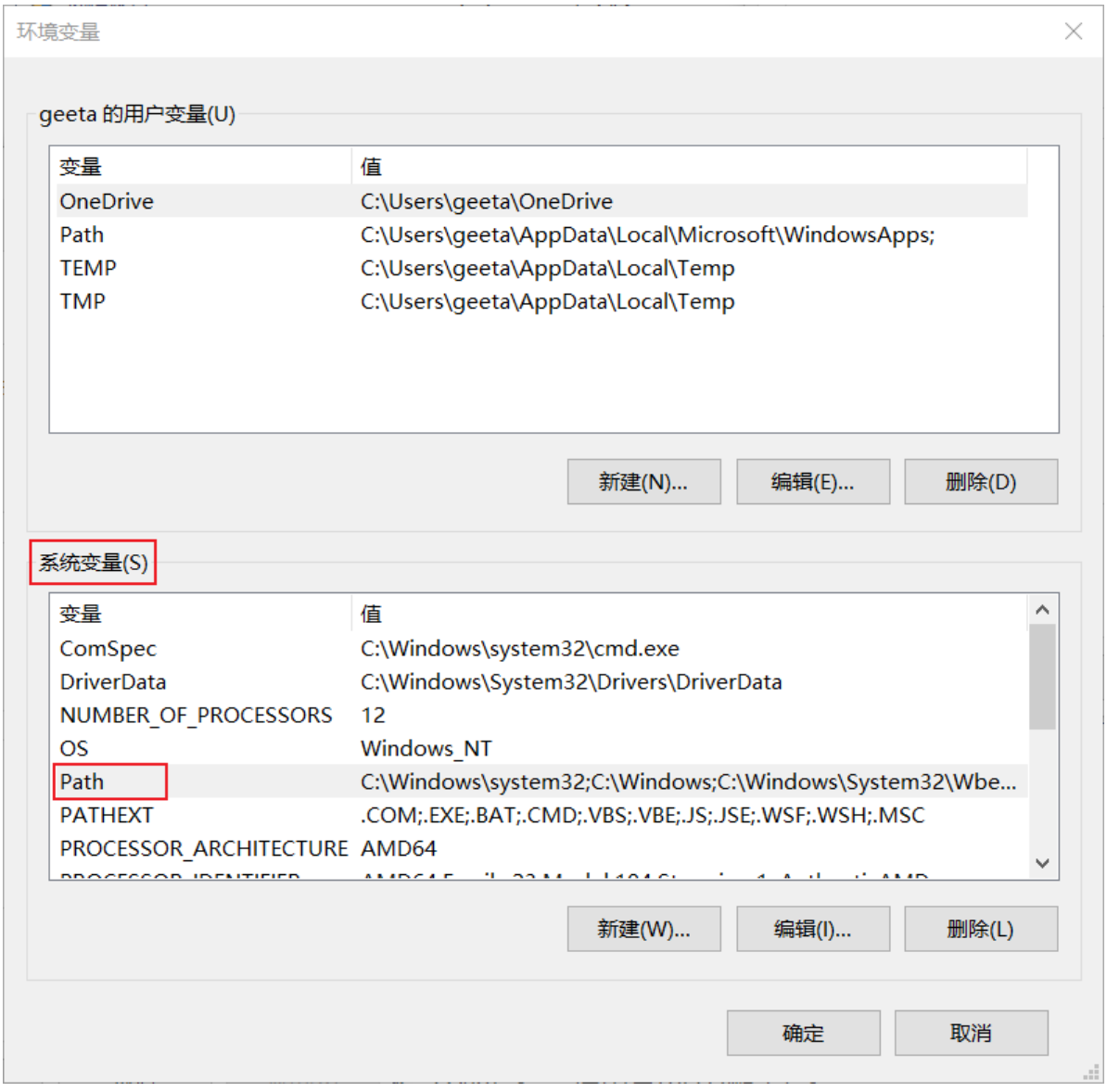
PATH环境变量的值就应该是javac和java工具所在的目录路径。

配置环境变量

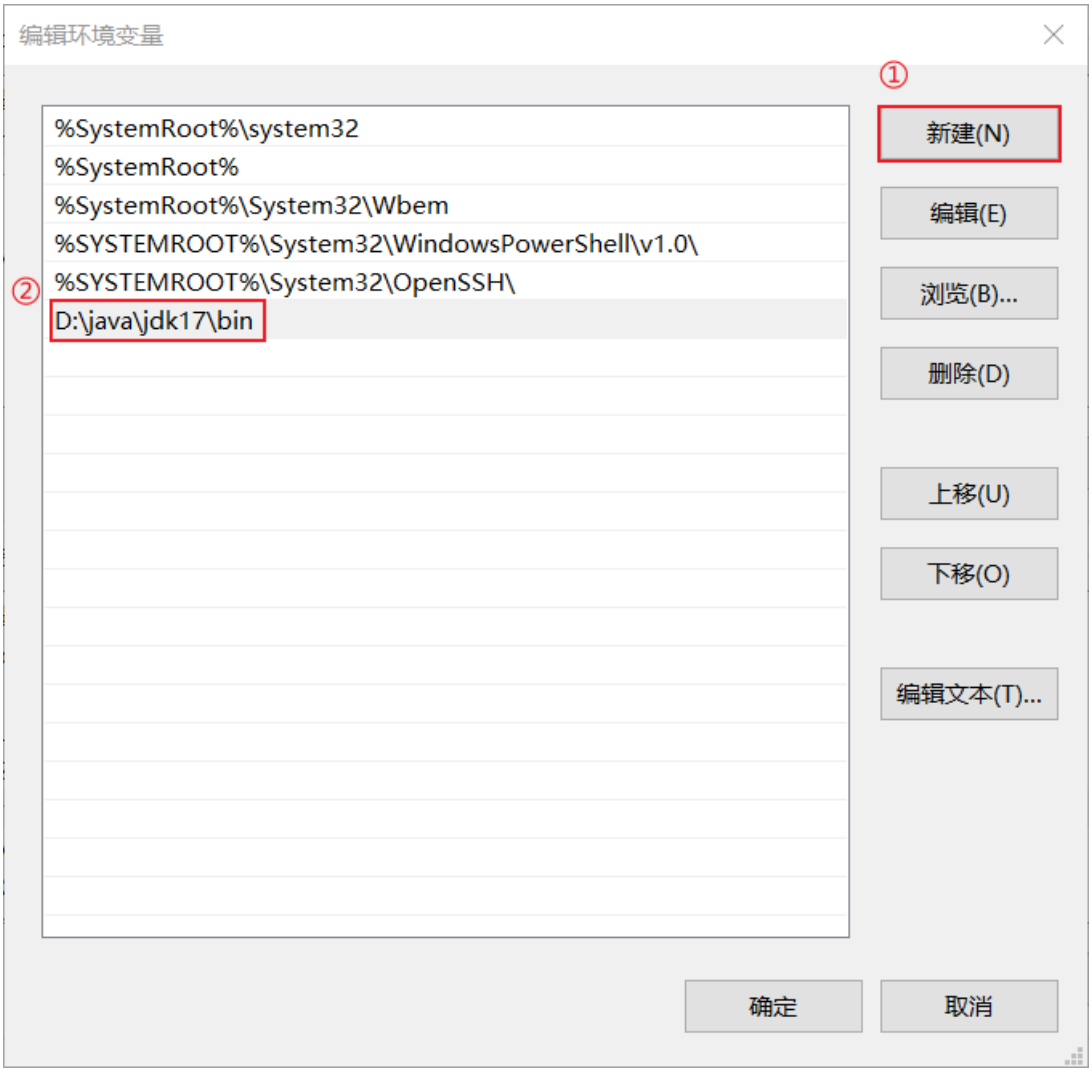
step 1：我的电脑 → 右键属性 → 高级系统设置 → 环境变量



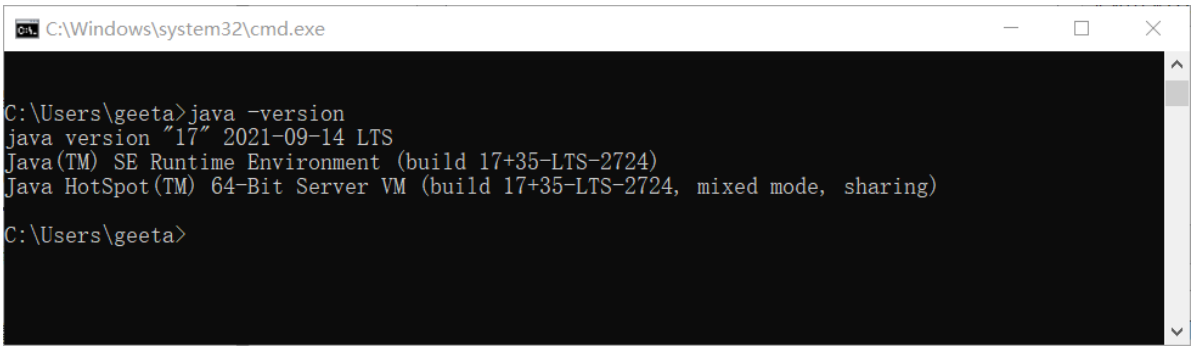
step 2：打开环境变量，找到系统环境变量中的PATH环境变量



step 3：把`D:\java\jdk17\bin`加入环境变量中



step 4：**重新打开命令行窗口**，在命令行输入 javac 或者 java 命令，也可以直接输入 java -version 看安装的jdk版本



如果看到上述界面说明PATH配置成功，此时就可以开发Java程序了。

常见问题：

- 忘记配置PATH环境变量
- PATH环境变量没有指向JDK下面的bin目录
- 出现javac不是内部或者外部命令的错误，原因是PATH配置错误

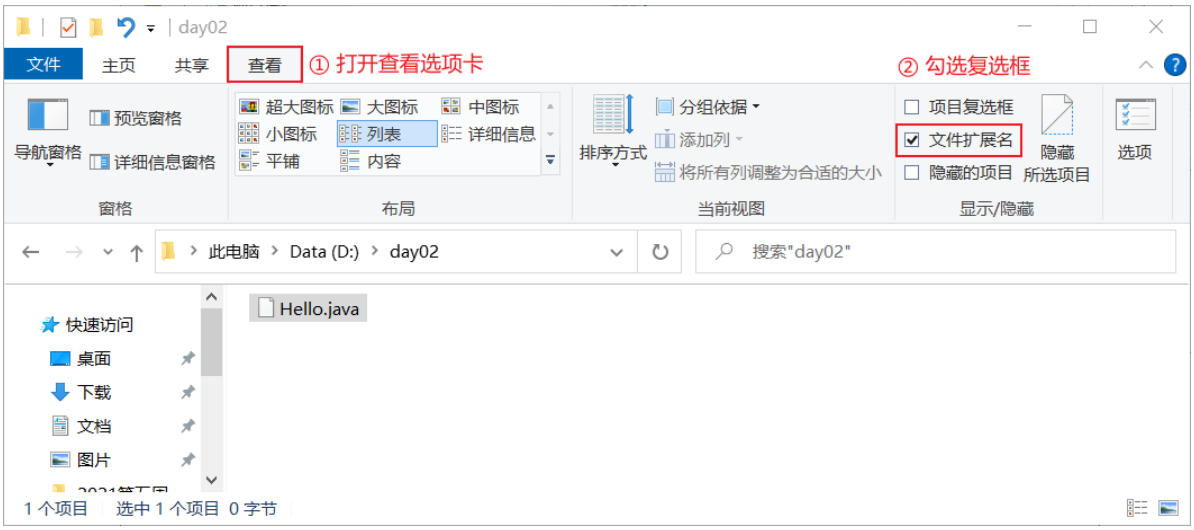
3、第一个Java程序（重点）

一般而言，编写的第一个程序，习惯叫做HelloWorld程序。

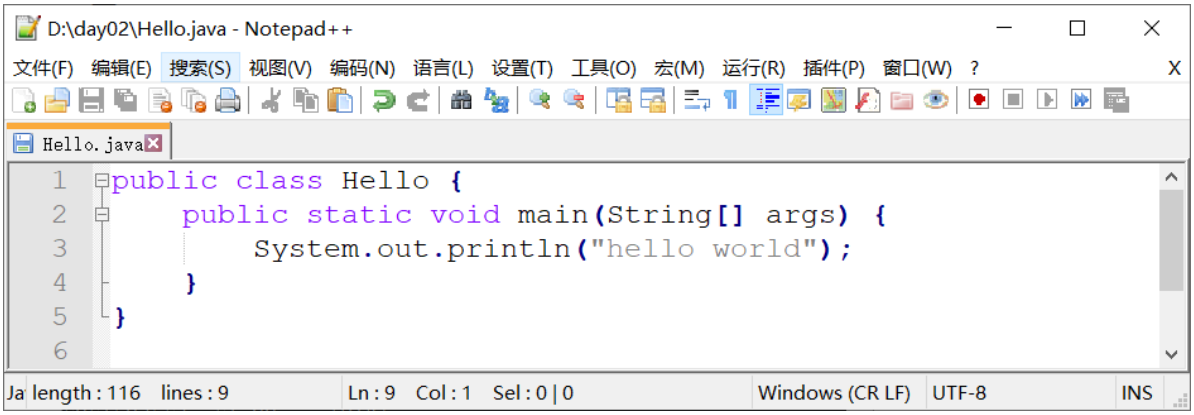
需求：在控制台（Win命令行）界面，输出：hello world (你好师姐)

3.1 HelloWorld程序开发（重点）

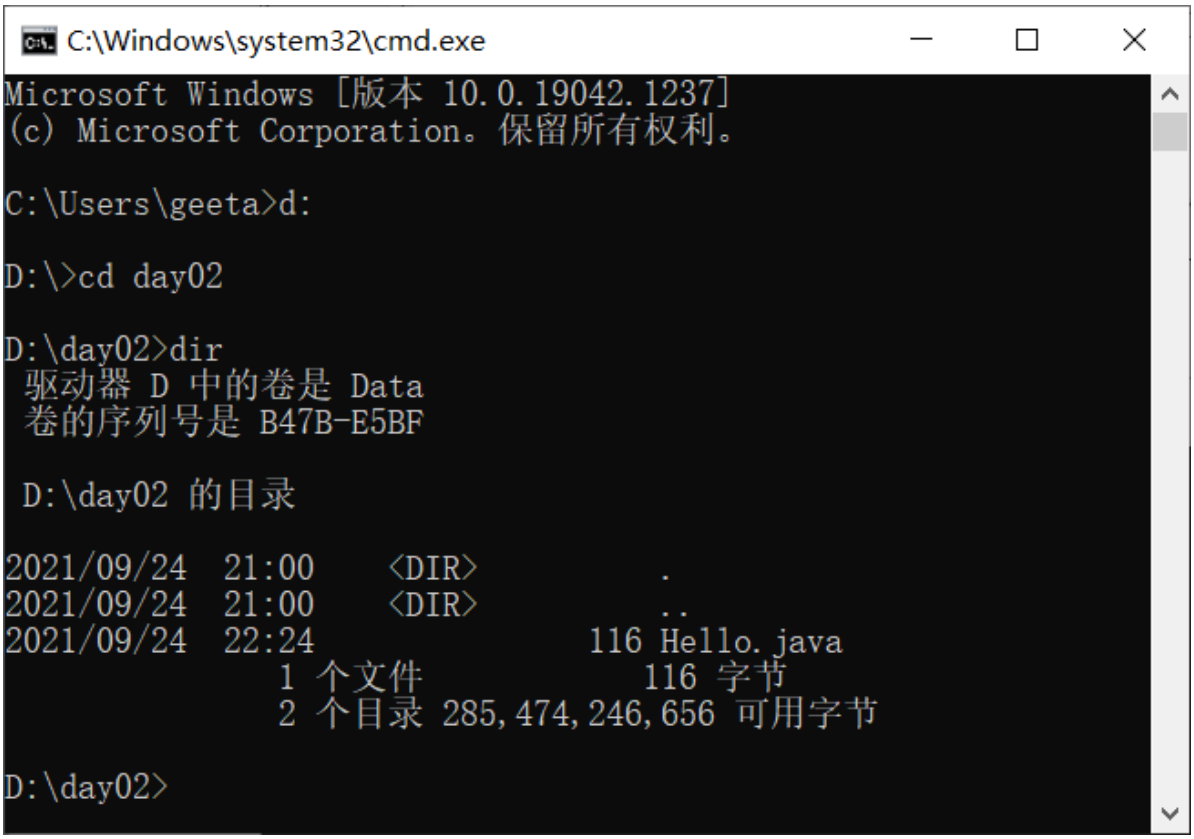
step 1：在开发盘(例如：G盘)创建一个文件夹，命名为day02，此时需要在文件夹选项中启用文件拓展名。在day02下新建文本文档，修改名称为Hello.java。



step 2：使用记事本或Editplus或**Notepad++**等文本编辑器，打开Hello.java文件，输入以下代码



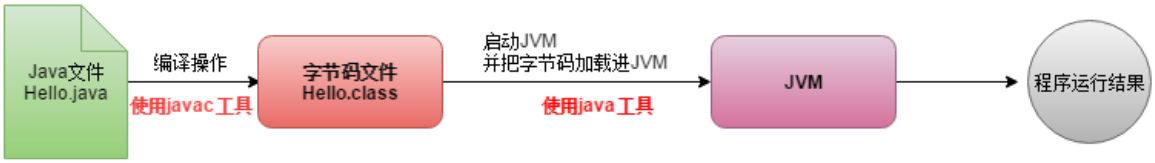
step 3：命令行进入当前源文件所在目录。



step 4：编译&运行

```
C:\Windows\system32\cmd.exe
D:\>cd day02

D:\day02>dir
 驱动器 D 中的卷是 Data
 卷的序列号是 B47B-E5BF

 D:\day02 的目录

2021/09/24  21:00    <DIR>          .
2021/09/24  21:00    <DIR>          ..
2021/09/24  22:24               116 Hello.java
               1 个文件            116 字节
               2 个目录 285,474,246,656 可用字节

D:\day02>javac Hello.java    编译源文件

D:\day02>java Hello  运行字节码文件
hello world

D:\day02>
```

3.2 Java的编译和运行机制（重点）



- 编写源文件（Java文件），源文件中包含源代码（Java代码）。
- 使用javac工具对源文件做编译操作，编译过程主要用于进行语法检查。

```
# 编译命令格式
javac 源文件.java
```

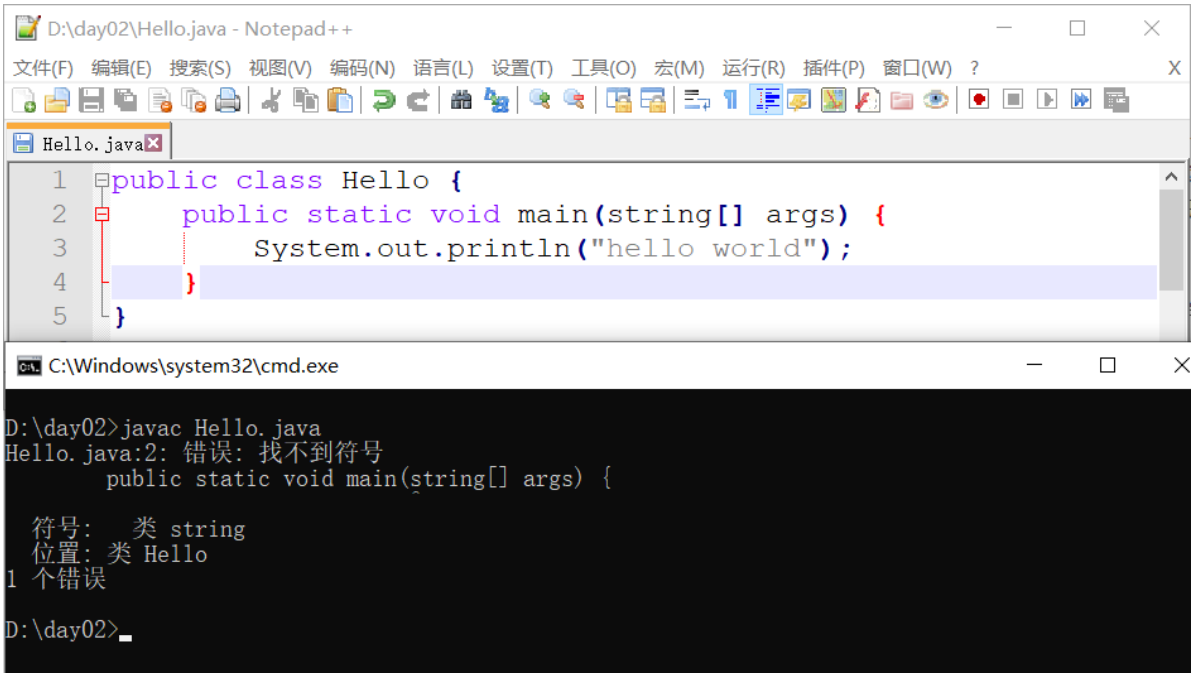
- 生成字节码后，使用java工具启动JVM，运行程序

```
# 运行字节码的命令格式
java 拥有主方法的类名
```

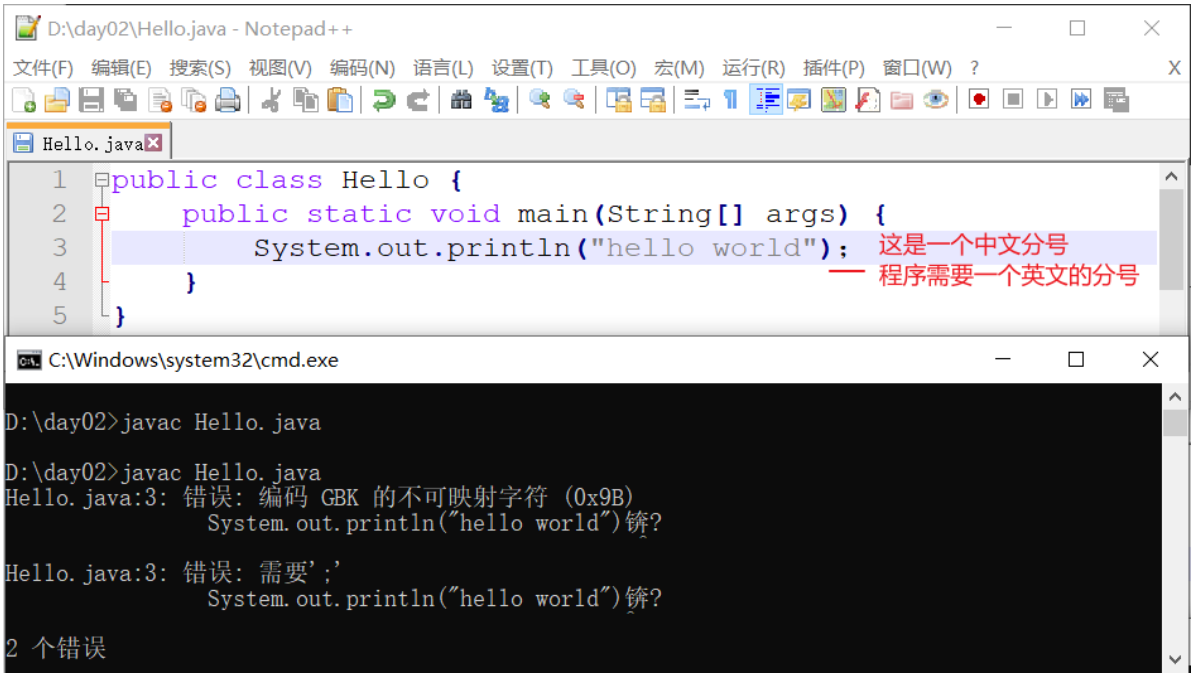
- 在控制台输出，显示结果

3.3 HelloWorld程序常见问题（了解）

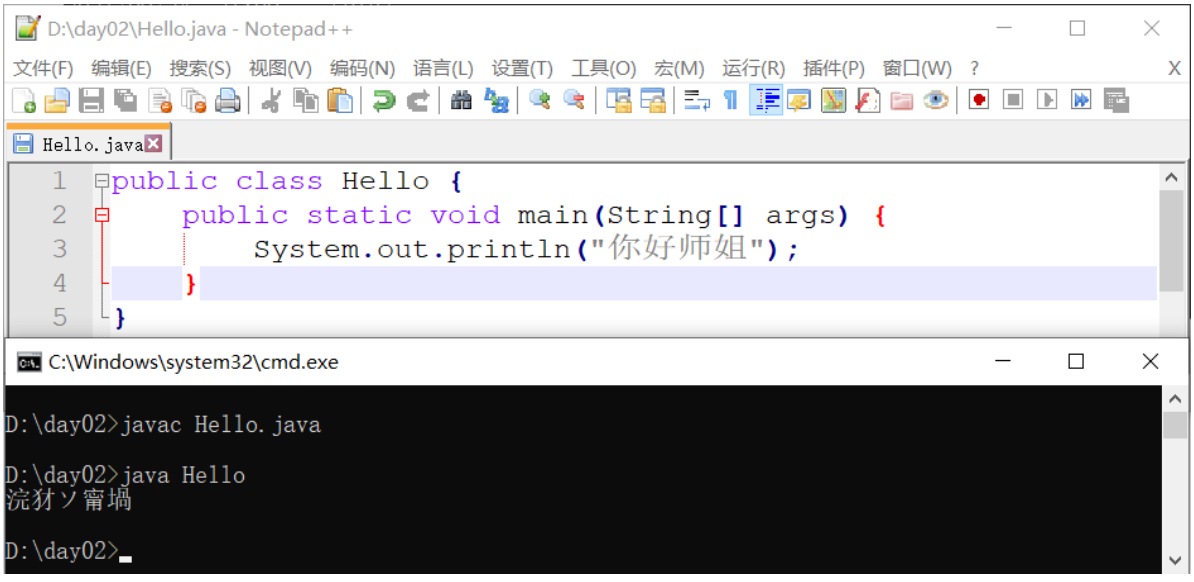
第一类：大小写问题。Java语言是严格区分大小写的



第二类：中英文输入法问题



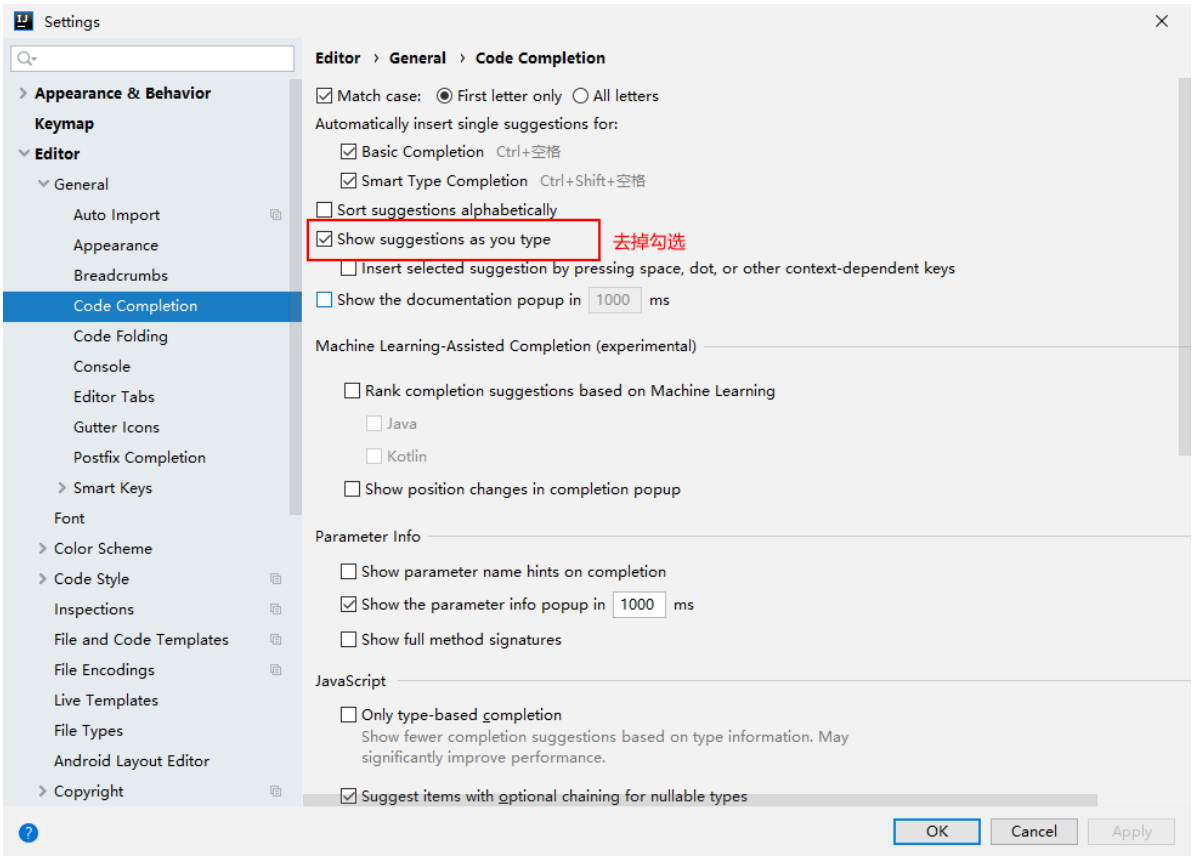
第三类：控制台输出中文问题



3.4 开发工具 - IDEA

IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。IDEA是JetBrains公司的产品，这家公司总部位于捷克共和国捷克共和国/418555)的首都布拉格，开发人员以严谨著称的东欧程序员为主。它的旗舰版本还支持HTML，CSS，PHP，MySQL，Python等。免费版只支持Java等少数语言。

为了形成良好的指感，早期学习建议大家不使用智能提示！！！



4、Java程序的基本语法（重点）

4.1 基本语法（必须记住）

- **Java程序以类(class) 组织的，创建Java程序时必须首先得创建一个类。**
- **若一个类必须运行，则必须拥有main方法（主方法）**,因为main方法是程序的入口。
- **Java语言严格区分大小写。**好比main和Main、system和System是完全不同的概念。
- 一个Java源文件里可以定义多个Java类，但其中最多只能有一个类被定义成public类。若源文件中包括了public类，源文件必须和该public类同名。
- 一个源文件中包含N个Java类时，成功编译后会生成N份字节码文件，即每个类都会生成一份单独的class文件，且字节码文件名和其对应的类名相同。

4.2 注释符号（了解）

Java的注释信息是给程序员看的，编译器(javac)在编译的时候会忽略掉源文件中的注释信息。

Java提供3种注释类型：

单行注释：// 内容，//后面这一行的内容被注释
多行注释：/* 内容 */；/* 和 */ 之间的所有内容被注释
文档注释：/** 内容 */；/** 和 */之间内容被注释，此外，还可以专门生成文档信息API

注意：多行注释之间彼此都不能交叉嵌套，以下就是错误的例子。

```
/*
  /* */
*/
```

因为 /* 会找距离自己最近的 */ 符号，组成一个注释语句块，上图中单独的*/符号就不能被编译器识别了。

标准的代码案例（必须这么去做）

后续代码的编写，要严格都要写需求、开发步骤、小结，以便初学者形成编程思维。

需求：在控制台输出内容：你好师姐

```
/*
需求：在控制台输出内容：你好师姐
开发步骤：
1：先定义一个类Hello
2：在Hello类中定义一个main方法
3：在main方法中使用系统输出打印一句话
4：编译和运行程序
*/

public class Hello {
  public static void main(String[] args) {
    System.out.println("你好师姐");
  }
}

/*
写完代码后，写下小结：我在写代码过程的得与失
1：如果不写main方法，程序没法运行，因为main方法是程序的入口。   mian
2：Java严格区分大小写，System不能写出system，其他也要注意。
3：巨坑，分号必须使用英文状态下的，不能使用中文(中文的；英文的;)
*/
```

4.3 关键字和保留字（了解）

关键字：在编程语言中事先定义的，有着特殊含义和用途的单词。

保留字：和关键字一样是编程语言事先定义好的，只是说现在暂时没有特殊的功能，但说不定以后某天会突然被赋予功能，因此被保留下来的单词。比如goto和const就是保留字。

abstract	do	implement	private	this
boolean	double	import	protected	throw
break	else	instanceof	public	throws
byte	extends	int	return	transient
case	false	interface	short	true
catch	final	long	static	try
char	fianlly	native	strictfp	void
class	float	new	super	volatile
continue	for	null	switch	while
default	if	package	enum	synchronized
assert				

小结：注意关键字和保留字都是由小写组成，关键字不要去记，我们学一个记一个。

注意：

- java 无 sizeof 、goto、 const 关键字
- 有人认为：true，false，null属于字面量（直接量），不属于关键字，无所谓，不要纠结这些学术性。

4.4 分隔符和标识符

4.4.1 语言分隔符（了解）

分号（;）：用于语句的分割，表示一句话结束，好比咱们使用的句号。
花括号（{}）：表示一个代码块，是一个整体，花括号要成对使用。
空格（ ）：把一整条语句分割成几段，空格的次数不限制，好比一句英文里单词都要分开写一样。
方括号（[]）：定义数组和访问数组元素时使用。
圆括号（()）：使用很广泛，具体用到细讲。
圆点（.）：类和对象访问它的成员时使用。

注意：**必须都是半角状态下的英文符号**，写代码时确保搜狗输入法是这个样子的。 

小技巧：写代码时，全部使用英文的符号，不要使用中文的符号，修改搜狗输入法配置。

4.4.2 标识符（记住）

在写代码的时候为了增强代码的可阅读性，我们会自定义很多名字。如：类名、方法名、变量名等。

在编程的世界里，我们把这种为了增强程序阅读性而自定义的名称，统称为标识符。

所有标识符的通用命名规则（必须记住）
[1].由字母、数字、下划线、$组成，但不能以数字开头
[2].大小写敏感
[3].不得使用java中的关键字和保留字

不同的标识符使用不同的命名规则（后续详细讲）：
类名:类名首字母大写，其他的字母小写(Hello),如果是多个单词，后面的单词首字母都大写(HelloWorld)。不用java内置的类名作为自己的类名。
变量名：未完待续
方法名：未完待续

5、数据类型、常量

5.1 常量（掌握）

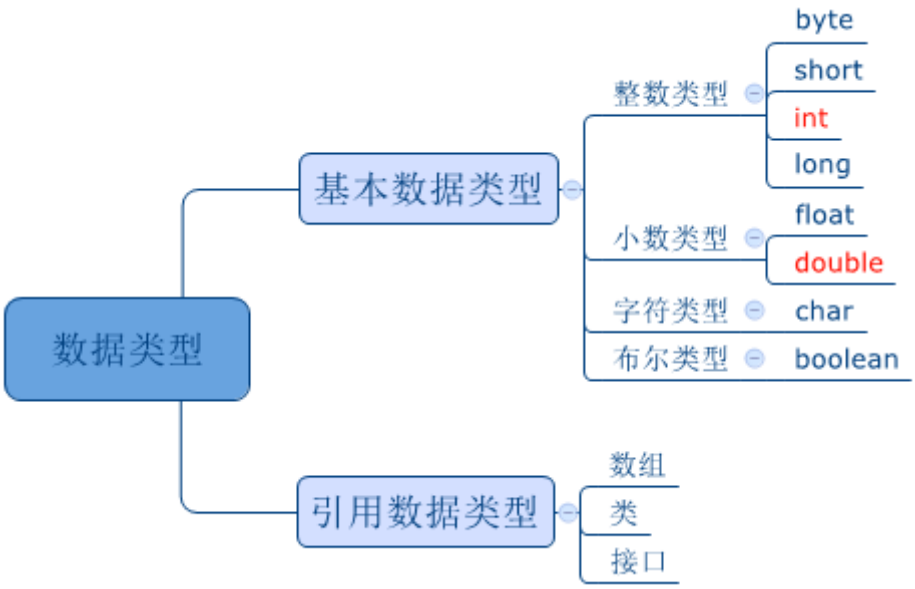
常量（const），程序运行过程中固定不变化的值。

常量分类:

- 字面量: 就表示直接给出的一个值（可以是整数、小数等），也有人称之为直接量。如整数常量1，2，3，小数常量3.14等。
- 使用final定义的变量（后讲）

5.2 数据类型（重点）

生活中，数据都是有类型这个概念的，比如张三18岁，18这个数字就是整型的，买了2.5斤菜，2.5就是小数类型的，在Java中每一个数据也有数据类型。



在计算机中，人为地把8个二进制位称为一个字节(byte)

```
(了解)
1个二进制位  可以存储0或1                可以存储2种状态
2个二进制位  可以存储00或01 10 11   可以存储4种状态
...
8个二进制位   可以存储2的8次方种状态，256种状态，如果每个状态我用来存储数字，可以存储256个数 => [0,255],如果还可以表示负数[-128,-1]，[0,127] => [-128,127]
```

8种基本数据数据类型的范围和**占内存大小**：

No.	数据类型	占位(字节)	数据范围
1	byte	1	[-128 , 127]
2	short	2	[-32768 , 32767]
3	**int**	**4**	[-2^{31} , 2^{31} - 1] ≈ [-21亿，21亿]
4	long	8	[-2^{63} , 2^{63}-1]
6	float	4	[-3.4×10^{38} , 3.4×10^{38}]
7	**double**	**8**	[-1.7×10^{308} , 1.7×10^{308}]
5	char	2	[0 , 2^{16}-1]
8	boolean	1位	true,false

提醒：开发者需要明确记住每个具体数据类型所占字节数（内存大小）。

整数类型常量

在 java 中，整型常量默认使用 int 类型来存储 。

```
public class IntTypeDemo {
    public static void main (String[] args) {
        // 控制台输出10，这个10是一个常量，默认使用int存储。
        System.out.println(10);

        // 控制台输出20，这个20如果想以long类型存储，需要加L
        System.out.println(20L);
    }
}
```

如果要存储long类型常量，要加L或者l，建议加L。**常用的整数类型是int和long**，byte和short基本不用。

小数类型常量

小数类型默认是double类型，如果要存储 float 类型常量，要加f或者F。

```
public class DoubleTypeDemo {
    public static void main (String[] args) {
        // 3.14是一个小数常量，默认使用double存储。
        System.out.println(3.14);

        // float类型常量,需要加F/f后缀
        System.out.println(3.14f);
    }
}
```

另外，实际开发过程中，如何选择这两种数据类型呢？，看小数的位数：

float类型精确到6-7位，double精确到15-16位，绝大部分应用程序都采用double类型。

```
// float 类型只能精确到6-7
System.out.println(3.141592653f);  // 3.1415927
```

注意：

- **小数类型在计算机中存储是不精确的**，所以，float，double 类型的数据不适合用在金融计算领域(不容许舍入误差)。如果需要精确数字计算，需要使用 BigDecimal 类 (后面学) 。

字符类型常量

Java中的字符表示Unicode（万国码）编码表中的每一个符号，每个符号使用单引号引起来，其中前128个符号和ASCII表相同，如下图。

ASCII 字符代码表 一

高四位 / 低四位		ASCII非打印控制字符										ASCII 打印字符															
		0000					0001					0010		0011		0100		0101		0110		0111					
		0					1					2		3		4		5		6		7					
		十进制	字符	ctrl	代码	字符解释	十进制	字符	ctrl	代码	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl			
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p				
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q				
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r				
0011	3	3	♥	^C	ETX	正文结束	19	‼	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s				
0100	4	4	♦	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t				
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u				
0110	6	6	♠	^F	ACK	确认	22	▬	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v				
0111	7	7	•	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w				
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x				
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y				
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z				
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{				
1100	C	12	♀	^L	FF	换页/新页	28	∟	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|				
1101	D	13	♪	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}				
1110	E	14	♫	^N	SO	移出	30	▲	^6	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~				
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	Back space			

这张表要记住的几个符号，A在码表的顺序是65，a在码表的顺序是97。

```
public class CharTypeDemo {
    public static void main (String[] args) {
        // 控制台输出字符A
        System.out.println('A');
    }
}
```

字符串类型（String）常量

所谓字符串就是多个字符合在一起，使用双引号引起来。例如：在开发中要打印一个用户的名字就可以使用字符串了，因为一个用户的名字是由多个字符构成的。

```
public class StringDemo {
    public static void main (String[] args) {
        System.out.println("叩丁狼");
    }
}
```

提醒：引用类型先不管，先记住String这个类，表示字符串类型就可以了，后续会学习。

布尔类型常量

boolean 类型只有两个值，true 和 false，分别表示对与错，在未来的开发中用于逻辑判断。

```
public class BooleanTypeDemo {
    public static void main (String[] args) {
        System.out.println(true);
    }
}
```

不同数据类型的常量：

- 整数常量，所有整数，如1、2、3、100、200等
- 小数常量，所有小数，如1.2、2.7、3.14等

- 字符常量，0~65535之间的整数或用单引号括起来的符号如，'A'、'a'、'龙' 等
- 布尔常量，只有true和false，分别表示对与错
- 字符串常量，使用双引号括起来的内容如："Will"、"wolfcode"等

需求：打印每一种数据类型的常量

```
public class TypeDemo {
    public static void main(String[] args) {
        // int类型常量
        System.out.println("十进制" + 20);
        System.out.println("二进制" + 0B00010100);
        System.out.println("八进制" + 024);
        System.out.println("十六进制" + 0x14);

        // long类型常量,使用L后缀
        System.out.println(20L);

        // float类型常量,使用F后缀
        System.out.println(3.14F);

        // double类型常量
        System.out.println(3.14);

        // char类型常量
        System.out.println('A');

        // boolean类型常量
        System.out.println(true);
        System.out.println(false);

        // String类型常量
        System.out.println("你好");
    }
}
```