# 第二章 Java概述

## Java技术体系

Java SE 标准版；Java EE 企业版；Java ME 迷你版

## Java特点

Oop 面向对象；

健壮性；

跨平台；（.class文件可以在多个系统下运行）

Java语言是解释性语言。（再被机器执行前有解释器进行解释再进行执行）

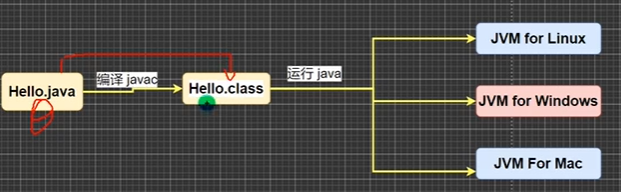
## Sublime安装

## Java运行机制

.java~（javac指令） .class~（java指令运行） JVM

不同的系统，Java虚拟机JVM也不一样，JVM屏蔽了运行系统的差异，包含在jdk中

.class称作字节码文件



## JRE运行环境

JRE=JVM+Java核心类库

## 环境变量path

向系统中添加默认查找目录

## DOS窗口快捷键

Win+R，cmd回车

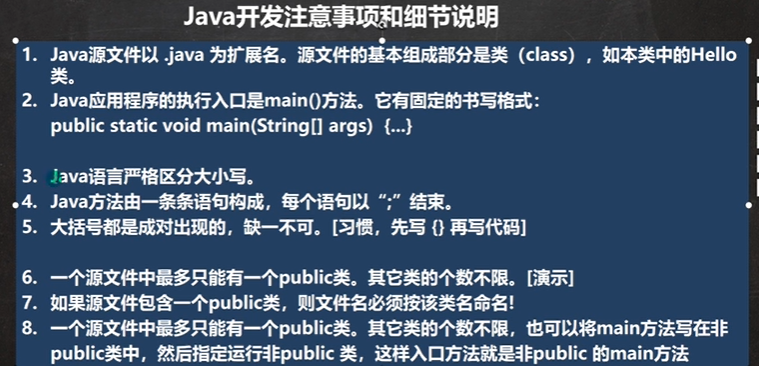
## 用户环境变量与通用环境变量

用户环境变量只为当前用户的环境变量设置

## 运行注意事项

在javac命令时需要写 文件名.java；在java时不需要写 文件名.class；而是直接写文件名即可

在内容有中文时把编码格式设置为GBK

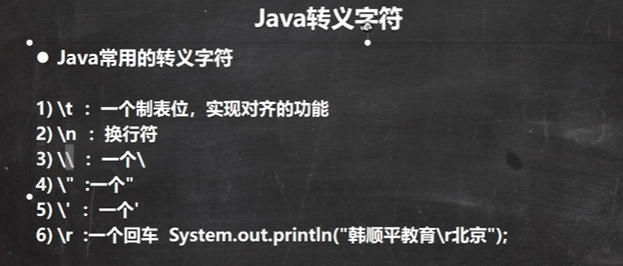


一个java文件中有public类，则文件名和类名要一致，否则会报错

Java文件可以包含多个类，在编译的时候，一个类会生成一个.class文件

在每个类中都可以有一个主方法，在编译后可以用 java 类名 分别调用

## 转义字符



（在DOS窗口中的tab键可以实现命令补全）

\r仅回车不会换行，所以一般会和\n连用

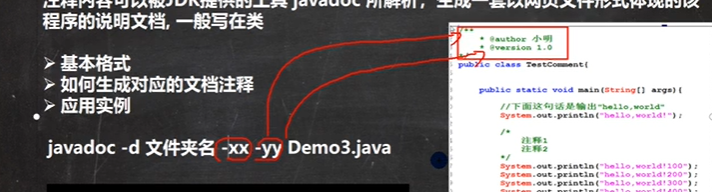
## 注释

// 单行注释

/\*…\*/ 多行注释

注释不可嵌套

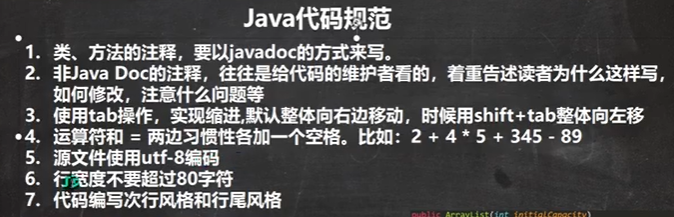
## 文档注释



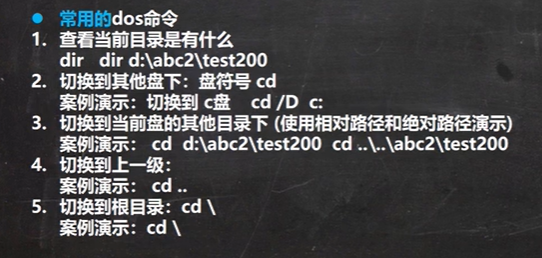
可以使用的javadoc标签

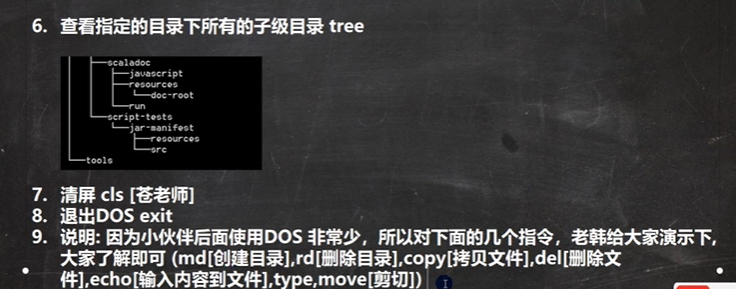


## Java代码规范



## DOS窗口命令





相对路径：从当前目录开始定位，指定的一个路径

绝对路径：从顶级目录开始定位，指定的一个路径

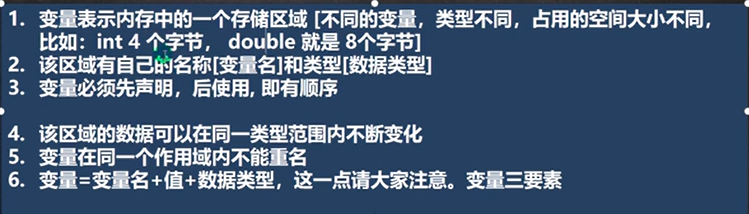
2021.04.23 Zxc

# 第三章 变量

## 变量使用步骤

声明变量→赋值→使用变量

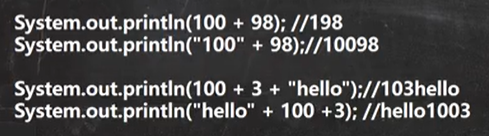
## 变量使用注意事项



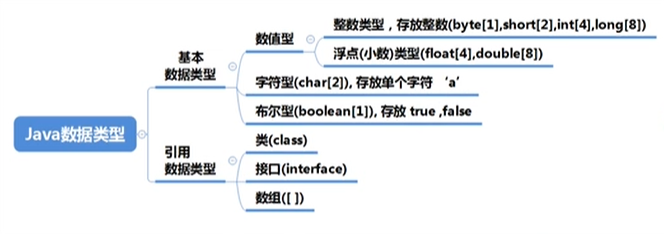
## 加号+的用法

当左右都是数值时，做加法运算

左右有一方为字符串，则做拼接运算

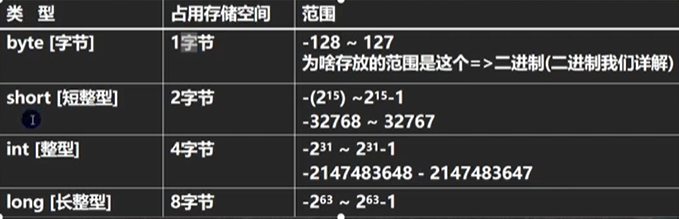


## 数据类型种类



注意：String不是基本数据类型

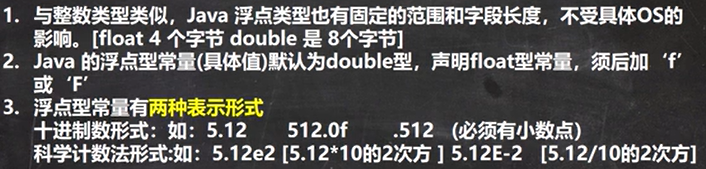
## 整型





## 浮点型

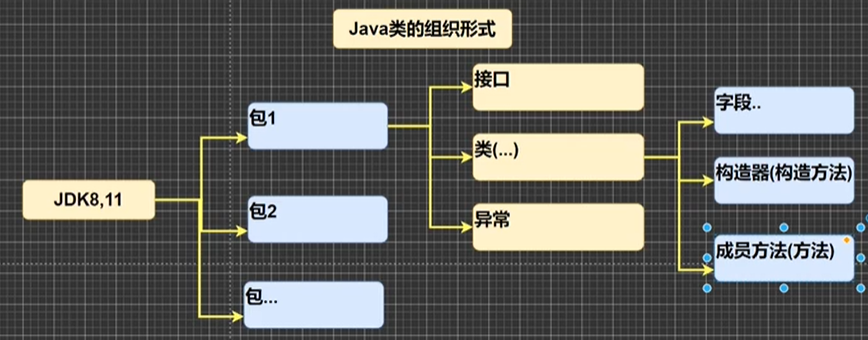




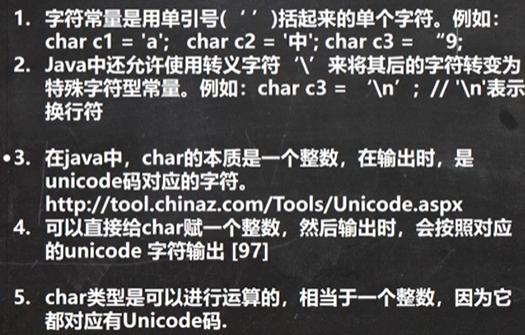
注意：不要对运算后的小数进行是否相等比较，因为小数是近似值，若要比较，应该用差值的绝对值在某个精度范围以内进行比较。也可以使用Math API。

在浮点型中System.out.print(2.0-1.1);时不会输出0.9，而是0.8999999999999，这是舍入误差造成的，因为在二进制中无法精确表示十分之一这个数，就像十进制中无法精确表示三分之一一样。

## Java类的组成形式



## 字符类型（char）



Char c = 99;会输出c，因为99是c的Unicode值；若将char类型值强制转向int型，则会变成Unicode值。Utf-8是在互联网上使用最广的一种Unicode的实现方式，英文字母和汉字都占2个字节

ASCII有128个值

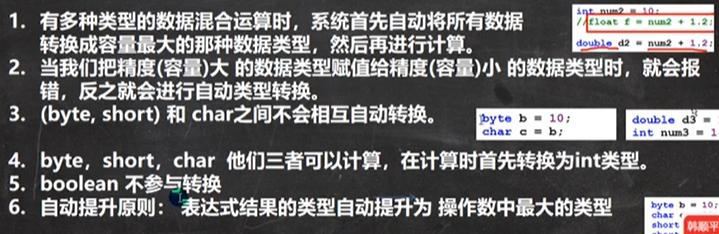
## 基本数据类型（primitive type）自动转换

精度小的类型自动转换为精度大的数据类型

Char→int→long→float→double

Byte→short→int→long→float→double

Byte、short和char之间不能自动转换



## 基本数据类型强制转换

将容量大的数据类型转换为容量小的数据类型，使用时加上强制转换符（），但可能造成精度降低或溢出

强制转换只对最近的符号有效



Char可以保存int常量值，但不能保存int变量值

Byte和short类型在运算时当做int处理

## 基本类型和String类型的互相转换

Primitive type → String ： 在后面加上+” ”;

String → primitive type ： Integer.parseInt();

Float.parseFloat();

Double.parseDouble();

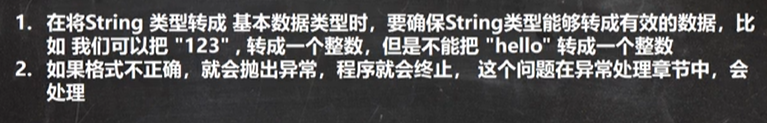
Long.parseLong();

Byte.parseByte();

Boolean.parseBoolean();

Short.parseShort();

注意事项：



2021.04.25 Zxc

# 第四章 运算符

## 算术运算符（Arithmetic operator）



int i = 1;

i = i++;

这个语句执行时会先开辟temp = i；然后i++；最后再i=temp；所以最终i的值还是1

int i = 1；

i = ++i；

这个语句则会i = i + 1;temp = i;i = temp;

## 关系运算符（Relational operator）



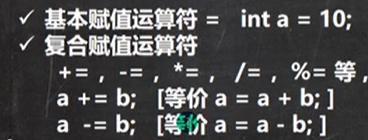
## 逻辑运算符（Logic operator）



&是逻辑与，&&是短路与，区别是，&&在对第一项判断时若为假，则不会再去判断第二项，而&是不管第一项的真假都会执行第二项

^是异或

## 赋值运算符（Assign operator）

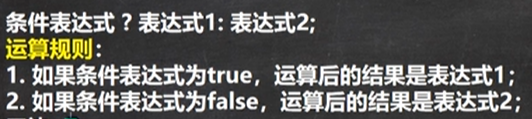


赋值运算符的左边只能是变量

复合赋值运算符会进行类型强制转换

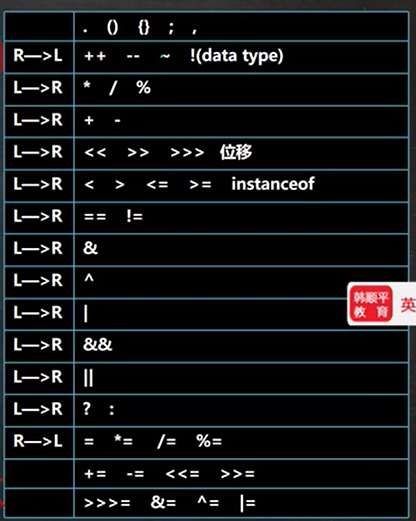
byte b = 1；b++；//b = (byte) (b + 1);

## 三元运算符（Ternary operator）



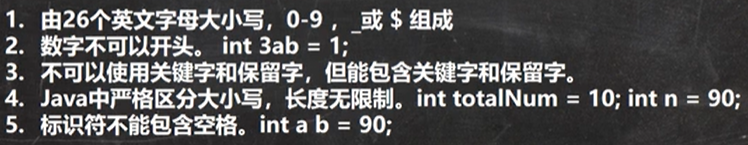
表达式1和表达式2要为可以赋给接受变量的类型（或可以自动转换）

## 运算符的优先级

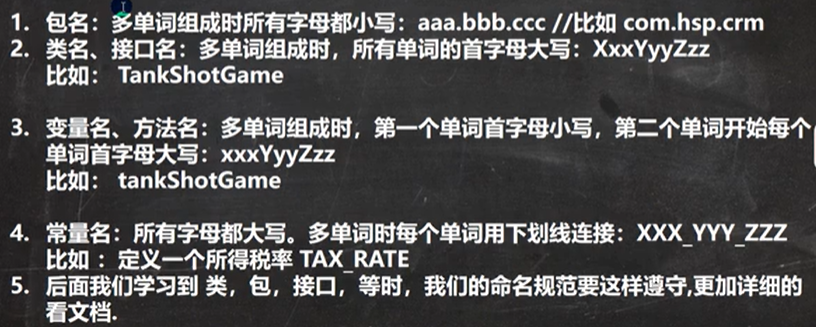


## 标识符的命名规则和规范

规则：

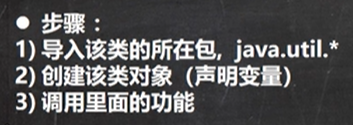


规范：



## 键盘输入语句

需要一个扫描器（对象），就是Scanner



import java.util.Scanner;//导包