一

Jdk：开发安装包，包含JRR

JRE:java运行环境，包含JVM

配置环境变量是为了去执行bin下面的javac.exe等指令，只有配置了路径才能找到指令，否则无法执行指令，除非切换到指令所在的目录

如：在 c> java.exe 报错 （因为没配置环境变量）

Cd 到 c：\dev\jdk1.8\bin>java.exe 可以执行

配置了环境变量是为了可以在任何目录下面都去执行javac.exe等指令

Path：windows执行命令时需要搜寻的路径

1. // 表示单行注释
2. /\*

表示多行注释 不可以嵌套使用

\*/

3.

/\*\*

文档注释 可以被javadoc解析

\*/

关键字,保留字

关键字：都是小写 class，if，private，int等

保留字：现在的java版本没使用，以后可能会用比如： goto,const

标识符：对变量，方法，类等命名的字符序列

1. 由26个英文大小写，0-9，\_或$组成
2. 数字不可以开头
3. 不能使用关键字和保留字
4. 区分大小写 如static（关键字）与Static（标识符）不同
5. 不含空格
6. 包名都小写
7. 类名，接口的首字母大写
8. 变量，方法第二个单词大写 常量全部大写

变量：包含变量类型，变量值和存储的值

数据类型：基本数据类型（八种），引用数据类型（类，接口，数组）

Byte为一个字节，为八位，8bit，每一位都有0,1二种表示，所以组合起来为2的八次方，所以范围是0~256 为了区分正负，所以范围是-128 ~ 127

Byte（字节）：-128 ~ 127 byte a1 = -128，byte b2 = 12 ，byte b3 = 128（超出范围）

Char（字符）: (1字符=2字节) char a =’a’ char=‘张’ char c = ‘bm’(“错误”) char里面只能有一个字符

Char c5 = ‘\n’ 换行

字符集：

内存或则硬盘都是存储的01010101等二进制

相当于对应法则 如：1. 中（A） --- > 字符集（65） --🡪变成电脑能懂的二进制（00000001）

电脑能懂的二进制--🡪字符集---🡪变成汉子或则数字符号等

代码编写java文件有中文 sout（“你好中国”），我们设置字符集为UTF-8，保存硬盘进去假设为1234

命令行 读取java文件采用GBK，找不到1234，就会出现乱码

自动类型提升：低范围数据类型可以自动转为高范围数据类型

Byte：一个字节（8位），char：二个字节（16位）

Byte,short,char(这三个变量做运算都只能用int接收)--🡪int --🡪long--🡪float-->double

Java做运算的时候，如果操作数据范围在int范围内，那么一律按int的空间内运算

整型常量默认时int类型 如 byte a = 1；int b = a + 3；（3 为int类型，所以需要用int接收）

浮点型默认为double 如：byte a = 1；double b = a + 3.2 (……)

String不能强转为基本数据类型

进制见的转换

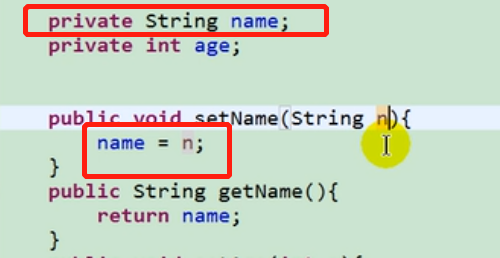
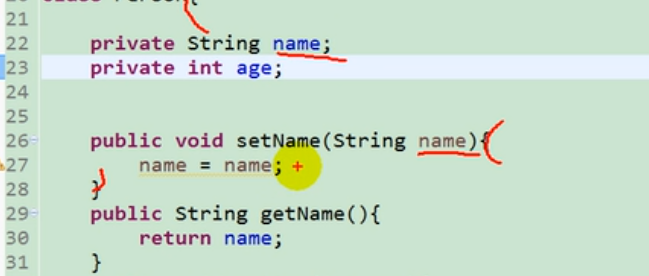
二进制：0,1 以0b或0B开头（计算机采用）

十进制：0-9，（平时工作计算中采用）

八进制：0-7以数字0开头

十六进制：0-9及A-F 以0x或0X开头

关键字：



1. 对于图2 由于name一个是全局变量，一个是在方面里面的，级别不一样，所以不报错
2. 对于图1 由于name从方法里面找不到，向上有全局变量可以找到，因此尽管不加this，name就是代表全局变量name，对于图2，由于name可以在方法里面找到，所以不加this就代表的是本方法里面的name，无法指向全局变量name（方法也是这样，因为方法名字一般来说不重载不相同，重载的情况下参数不一样，所以在类里面调用方法不需要加this）

This表示当前类的对象（A a = new A（） this相当于a）

<https://www.bilibili.com/video/av50420483/?p=217>

<https://www.bilibili.com/video/av50420483/?p=48>