**简单的DOS命令**

cd 改变子目录：cd.. 返回上一级目录。 cd\ 返回根目录。 cd+空格+目录 到达指定的目录。

md 在当前目录下创建一个子目录。用法：md +空格+子目录名。

rd 删除指定空目录，用法与md类似。

del 删除指定的文件，用法：del+空格+文件名。del+空格+目录，表示删除该目录下所有文件。del+空格+\*+.+文件后缀（如txt）表示删除当前目录下指定的后缀的文件（例中，删除所有txt文件）。

exit 退出DOS命令行。

Java环境变量配置完成后，使用DOS命令行的方式完成一个程序的编译和运行。首先创建一个记事本，编写如下内容：

public class Demo

{

public static void main(String[] args)

{

System.out.println("hello world");

}

}

并保存为后缀为.java的文件。然后，打开命令提示符界面，通过cd的命令到达刚才完成的java文件的目录下，然后输入javac+空格+文件名.java，实现文件的编译，这是文件下会出现一个名为Demo.class的文件（上面代码首行是否添加public有无是由其后的名称（此处为Demo）与文件名是否一致所决定的）。然后输入java+空格+文件名,便可以得到输出结果。

**使用IDEA编写hello world程序。**

首先有关IDEA的知识：

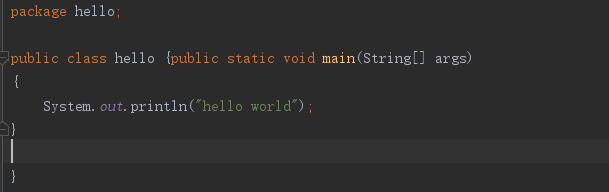
**清除缓存和索引：**File->Invalidate / Restart这样清除较干净，本质也就是去删除 C 盘下的 system 目录下的对应的文件。如果遇到了因为索引、缓存坏了以至于项目打不开，可以直接删除 system 目录来解决。

**IDEA的编译方式：**1. Compile：对选定的目标（Java 类文件），进行强制性编译，不管目标是否是被修改过。2. Rebuild：对选定的目标（Project），进行强制性编译，不管目标是否是被修改过，由于 Rebuild 的目标 只有 Project，所以 Rebuild 每次花的时间会比较长。3. Make：使用最多的编译操作。对选定的目标（Project 或 Module）进行编译，但只编译有修改过的文 件，没有修改过的文件不会编译，这样平时开发大型项目才不会浪费时间在编译过程中。

**Language level：**限定项目编译检查时最低要求的 JDK 特性。例如当我们项目使用的是 JDK 8，但是代码却没有使用 JDK 8 的新特性，最多使用了 JDK 7 的特性的时 候我们可以选择 7 - Diamonds，ARM，multi-catch etc. 。查看alt+shift+Ctrl+s->project，可以看到有project language level。

接下来新建一个项目，命名以及存放地址选择完毕后，在右侧目录的src下新建一个package，

然后在该package下新建一个class，命名完成后既可编写代码



之后build，run。

代码的详细介绍：

public class Demo

{

public static void main(String[] args)

{

System.out.println("hello world");

}

}

Public为公开给其他类存取，class进行类声明，Demo为类名称，void代表没有返回值

Main表示方法的名称，System.out.println为标准输出其后双引号内容。其中print和println的区别在于后者会在最后换行，类似c中的\n。

Java中所有东西都属于某个类，建立的源文件为后缀为.java的源文件，编译后会产生后缀为.class的新的类文件。真正被执行的是类。其中的main（）为程序的起点。

**boolean：**boolean是关键字，表示就像c和c++中的bool一样。但是java中的boolean只能是true和false，而不能像c和c++中的可以bool flag =1；这样赋值。而只能像boolean b = true或者boolean b = false;这样赋值。而且Boolean与boolean不同。

Java中的if-else和while：

**1.**class IfTest {

public static void main (String[] args) {

int x = 3; if (x == 3) { System.out.println(“x must be 3”);

}

System.out.println(“This runs no matter what”); }

}

**2**.class IfTest2 {

public static void main (String[] args) {

int x = 2; if (x == 3) { System.out.println(“x must be 3”);

} else {

System.out.println(“x is NOT 3”);

}

System.out.println(“This runs no matter what”);

}

}

**3.**public class Loopy {

public static void main (String[] args) {

int x = 1; System.out.println(“Before the Loop”);

while (x < 4) {

System.out.println(“In the loop”);

System.out.println(“Value of x is ” + x); x = x + 1;

}

System.out.println(“This is after the loop”);

}

}

除了上面的格式外，其他地方与C语言中if-else和while的用法类似

Switch用法：能用于switch判断的类型有：byte、short、int、char、String。

switch (ch) {

case 'B':

System.out.println("case one");

break;

case 'A':

System.out.println("case two");

break;

case 'C':

System.out.println("case three");

break;

}

case语句中少写了break，编译不会报错，但是会一直执行之后所有case条件下的语句，并不再进行判断，直到default语句

break的用法：break的作用是跳出当前循环块（for、while、do while）或程序块（switch）在程序块中的作用是中断和下一个条件的比较。continue用于结束循环体中其后语句的执行，并跳回循环程序块的开头执行下一次循环，而不是立刻循环体。例：

**public** **static** **void** main (String[] args) {

**int** j = 3;

**for** (**int** i = 0; i < 5; i++) {

**if** (i == j)  {

**continue**;

 }

System.out.println("i = " + i);

}

System.out.pintln("循环结束");

}

操作符问题：

+，-，\*，/，++，--，%功能与c大致相同，在java中+号还有链接前后字符串的作用

**int** a = 3;

**int** d = 1;

System.out.println("int /int :a/d="+a/d);

则输出的是3。另外不同精度的转换规则为byte 、short、int、long、float、double，从低精度到高精度自动转换，但高到低需要强制转换，转换方法和c中一致，在变量前加（数据类型）。

关系运算符：

关系运算符用于比较两个数值之间的大小，其运算结果为一个逻辑类型(boolean布尔类型)的数值。即true或者false。boolean类型只能比较相等和不相等，不能比较大小

System.out.println("9.5<8 :"+(9.5<8));

输出为9.5<8 :false。