用CD命令

如果要看文件夹里面有哪些东西,就输入dir  
因为cd只是进入文件夹,所以要和dir 配合着使用  
注意CD 为命令提示 后要有空格

切换盘符 输入“D:”  
就能切换到D盘  
进入D盘A文件夹 输入：cd A 回车  
就进入A文件夹，显示为 D:\A

**cd.. 返回上一层**



super表示当前对象的父类对象的一个引用

super可以调用当前对象的父类对象的方法

1. **class** cam0
2. {
3. **double** size=10;
4. **void** printsize()
5. {
6. System.out.println(size);
7. }
8. }
9. **class** cam1 **extends** cam0
10. {
11. **double** size=20;
12. **void** printsize()
13. {
14. System.out.println(size);
15. System.out.println(**super**.size);
16. **super**.printsize();
17. }
18. }
19. **public** **class** cam2
20. {
21. **public** **static** **void** main(String args[])
22. {
23. cam1 m=**new** cam1();
24. m.printsize();
25. }
26. }

输出为：

1. 20.0
2. 10.0
3. 10.0

**static 方法可以直接以 类名.方法名（）来调用，无须创建对象；**

**static成员变量被所有对象公用，也可以直接以类名.方法名（）来调用，无须创建对象；**

**public stati c final 为全局常量**

java中创建一个Dog类的对象dog， Dog dog = new Dog();这段代码可以分为两部分：Dog dog；  
dog = new Dog()；  
其中Dog dog；是我们声明了一个引用数据类型的变量，但是这个时候，只在栈内存中开辟了一个空间存放这个变量。而这个变量此时没有任何指向。但规定它只能指向Dog类的对象。

而dog = new Dog（）； 的意思是我们实例化了一个对象，这个时候在堆内存中开辟了一片空间存放这个对象，并让之前声明的引用数据类型变量指向了这个堆内存空间的首地址。

String a="abc"，这行代码被执行的时候，JAVA虚拟机首先在字符串池中查找是否已经存在了值为"abc"的这么一个对象。如果有，则不再创建新的对象，将使用串池里原来的那个内存，直接返回已存在对象的引用，而不会重新分配内存；如果没有，则先创建这个对象，然后把它加入到字符串池中，再将它的引用返回

String s=new String("abc")，不管串池里有没有"abc"，它都会在堆中重新分配一块内存，定义一个新的对象。

两个String对象==比较地址，s1.equal(s2)比较字符串内容

**synchronized(obj)**

**{**

**while(!condition)**

**{**

**obj.wait();**

**}**

**obj.doSomething();**

**}**

**在另一线程B中，如果B更改了某些条件，使得线程A的condition条件满足了，就可以唤醒线程A**

**synchronized(obj)**

**{**

**condition = true;**

**obj.notify();**

**}**

**//且wait()语句，必须try｛｝catch（Exception e）｛｝**

try{}

catch(){}

中try后面的{}就是一个块级作用域，所以内部定义的变量是局部变量

try

{

double result=2/3;

}

catch(Exception e)

{

System.out.println(e.getMessage());

}

System.out.println( "result is: "+result); //出错，超出了result的作用域

API:

创建一个将数据写入指定文件的输出数据流：若找不到文件，则先新建之

FileInputStream inFile=new FileInputStream("c:/Users/myth/Desktop/persons.txt");

FileOutputStream

Vector , elements()，返回各项枚举至Enumeration；addElement(项)

Hashtable的keys(),返回键的枚举至Enumeration；elements()返回值的枚举;put(Object键,Object值) ；get(Object key)返回键对应的值

Properties的propertyNames()，返回键的枚举至Enumeration；put(String键,String值); setProperty(String key, String value)、getProperty(键)；load(FileInputStream), store(FileOutputStream, String comments)

将属性可写入输出流store()，可从输入流中读属性load()

@Override是覆盖的意思，这是jdk自带的一个注解。表示该方法是继续过来或者实现的方法，如果加了该注解，它的父类或者实现的接口中没有该方法，则ide会报错。@Override置于要覆盖父类的方法定义前：@Override public boolean equals(){}

要成功的从Hashtable中检索数据，用作关键字的对象必须正确覆盖了Object.hashCode()方法和Object.equals()方法；因为源码get()方法中if ((e.hash == hash) && e.key.equals(key))才返回value

public int **read**(byte[] b) throws [IOException](mk:@MSITStore:C:\Users\myth\Desktop\JDK_API_1_6_zh_CN.CHM::/java/io/IOException.html)，从文件指针所指位置读，而不是从文件开头。返回的是读进数组的字节量。如果读之前文件指针已指向文件末尾，则返回-1.

字符串“hello文化苦旅”的长度为："+str.length() 9

在文件中写入：

String str="hello文化苦旅";

raf.write(str.getBytes()); //raf为RandomAccessFile类的实例

从文件中读出：

raf.seek(0);

byte buf[]=new byte[100];

int length=raf.read(buf);

System.out.println(new String(buf,0,length));

Java中的字符是Unicode编码

PipedInputStream中实际是用了一个1024字节固定大小的循环缓冲区。写入PipedOutputStream的数据实际上保存到对应的 PipedInputStream的内部缓冲区。从PipedInputStream执行读操作时，读取的数据实际上来自这个内部缓冲区。如果对应的 PipedInputStream输入缓冲区已满，任何企图写入PipedOutputStream的线程都将被阻塞。而且这个写操作线程将一直阻塞，直至出现读取PipedInputStream的操作从缓冲区删除数据。这也就是说往PipedOutputStream写数据的线程Send若是和从PipedInputStream读数据的线程Receive是同一个线程的话，那么一旦Send线程发送数据过多（大于1024字节），它就会被阻塞，这就直接导致接受数据的线程阻塞而无法工作（因为是同一个线程嘛），那么这就是一个典型的死锁现象，这也就是为什么javadoc中关于这两个类的使用时告诉大家要在多线程环境下使用的原因了。

fw.write(arr,0,num); //FileWrite使用了缓冲区，不关闭，运行后demo2.txt文件中未被写入任何内容arr为byte型数组

File f = new File("d:" + File.separator+"test.txt");  
    InputStream in=new FileInputStream(f);  
   byte[] b=new byte[(int) f.length()]; //根据文件大小，指定字节数组长度

字节流在操作的时候本身是不会用到缓冲区（内存）的，是与文件本身直接操作的，而字符流在操作的时候是使用到缓冲区的

字节流在操作文件时，即使不关闭资源（close方法），文件也能输出，但是如果字符流不使用close方法的话，则不会输出任何内容，说明字符流用的是缓冲区，并且可以使用flush方法强制进行刷新缓冲区，这时才能在不close的情况下输出内容

ASCII（American Standard Code for Information Interchange，美国标准信息交换代码）

涓：UTF-8码：0xE6B693 GBK：0xE4B8 Unicode：6D93

方法中内部类不能是static类型

一般的内部类可以是static类型

复选框Checkbox:

--🡪ItemEvent----🡪ItemListener---🡪(Checkbox)e.getItemSelectable();

CheckboxGroup cbg.getSelectedCheckbox()

菜单项MenuItem：

--🡪ActionEvent----🡪ActionListener---🡪String e.getActionCommand();得到菜单项构造函数MenuItem(String label)中指定的标签

鼠标Frame上画画：

--🡪MouseEvent----🡪MouseListener---🡪(Frame) e.getComponent()

e.getID()=MouseEvent.MOUSE\_PRESSED

按钮：

--🡪ActionEvent----🡪ActionListener---🡪(Button) e.getComponent()--🡪(Frame）btn.getParent()得到按钮的父容器Frame

C++ ：源程序.cpp文件-----🡪编译为.obj目标文件-------🡪.exe可执行文件

汇编：源程序.ASM文件-----🡪汇编程序（MASM.EXE）对源程序进行汇编，汇编为.OBJ目标文件-------🡪用程序（LINK.EXE）处理.obj，形成可执行文件（.EXE）

**类加载:**Java命令的作用是启动虚拟机，虚拟机通过输入流，从磁盘上将字节码文件(.class文件)中的内容读入虚拟机，并保存起来的过程就是类加载。

for（ 1 ：2 ）用法：1必须是 类型，2必须是 对象名：集合名

for(string item:buys)等价于

for(int i=0; i<buys.lenth(); i++){

String item = buys[i];

}

class Animal

{

int age=30;

}

public class Dog extends Animal

{

int age=20;

public static void main(String []args )

{

Dog dog=new Dog();

System.out.println(dog.age);

}

}

子类成员变量与父类某个非private成员变量同名，则dog.age包含两个域，一个存父类age值，一个存子类age值，但“dog.age”格式只能访问子类age值。

如果父类非private成员变量未与子类某个成员变量同名，则 dog.age仅包含一个域，即父类的age值

