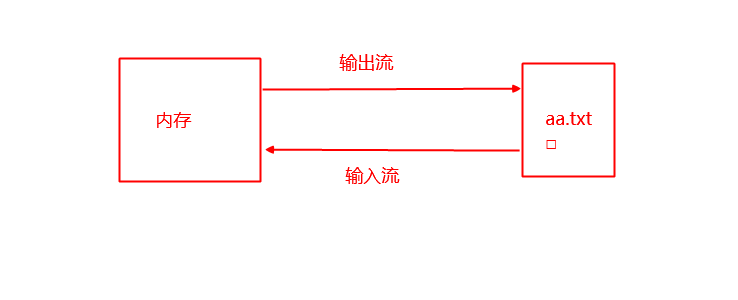
Java基础-----IO流

基本概念

文 件：文件是数据源(保存数据的地方)的一种，像我们经常使用的word,txt,java都是文件，文件最主要的作用就是保存数据，它可以保存文字、图片或声音、视频等

文件流：文件在程序中是以流的形式操作的

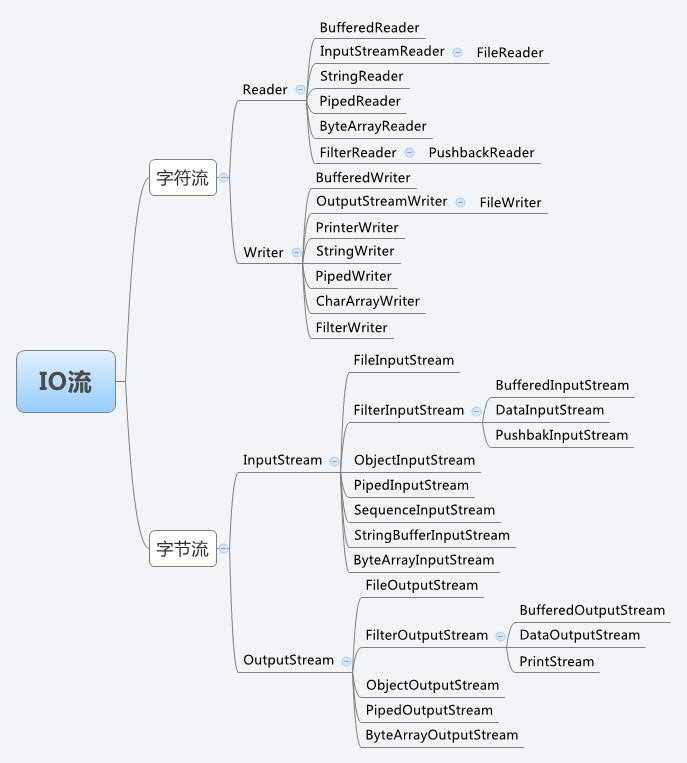
cc

流：数据在数据源和程序(内存)之间的路径

输入流：数据从 数据源(文件) 到 程序(内存) 之间的路径

输出流：数据从 程序(内存) 到 数据源(文件) 之间的路径

IO流一览表



Java中的文件流分两种，

1. 字节流：以字节为单为进行读写的流，可以操作二进制文件(图片、声音、影象…)及任何类型的文件
2. 字符流：不可以操作二进制文件，可以用于读写文本等内容为字符类型文件

输 入 输 出

字节流 InputStream OutputStream

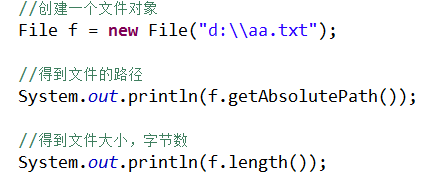
字符流 Reader Writer

注：InputStream及OutputStream 为字节输入流和字节输出流的父类，它们都是抽象类，不能被实例化，所以使用时，使用它们的子类来进行字节的输入输出操作

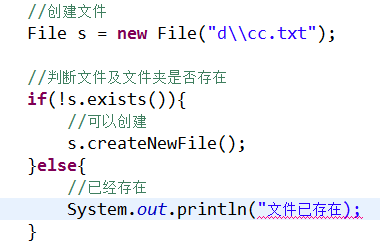
Rread和Write也是抽象类，故使用时也以它们的子类来实现字符文件的输入输出

使用：

File类的基本用法



创建一个文件，如果没有则创建



创建文件夹



取出文件夹下的所有文件

If(ff.isDirectory()){

File files[ ] = ff.listFiles;

}

使用io流来操作文件

案例一：读取文件(了解fileinputStream流)，输出到控制台

Eg:

File file = **new** File("d:\\aa.txt");

**try** {

FileInputStream fis = **new** FileInputStream(file);

//定义一个字节数组

**byte** []b = **new** **byte**[1024];

**int** n = 0;//实际读取到的字节数

**while**((n = fis.read(b))!=-1){

//把字节转成String

String ss = **new** String(b,0,n);

System.*out*.println(ss);

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

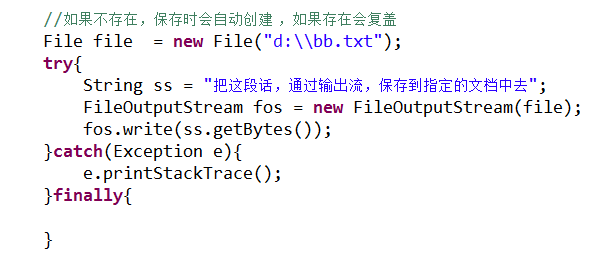
e.printStackTrace();

} **finally**{

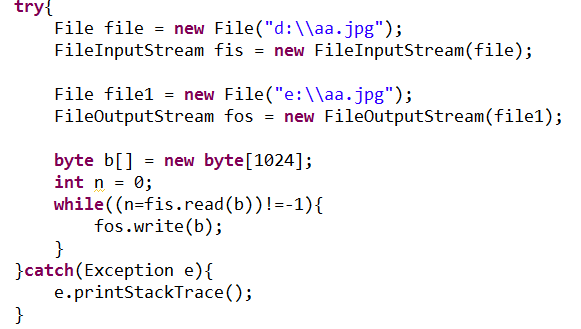
//关闭资源

}

案例二：从键盘输入，保存到文件中(演示fileoutPut流)



案例三：把d盘图片aa.jpg通过io流的读入和输出操作，保存到e盘



字符流：

FileReader FileWriter

字符流每次读取一个字符（二个字节）

案例：读取一个文件并写入到另一个文件中char[ ]用来中转

FileReader fr = **null**;

FileWriter fw = **null**;

**try** {

fr = **new** FileReader("d:\\aa.txt");

fw = **new** FileWriter("e:\\bb.txt");

**char** ss[] = **new** **char**[1024];

**int** n = 0;

**while**((n = fr.read(ss))!=-1){

fw.write(ss,0,n);

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally**{

**try** {

fr.close();

fw.close();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

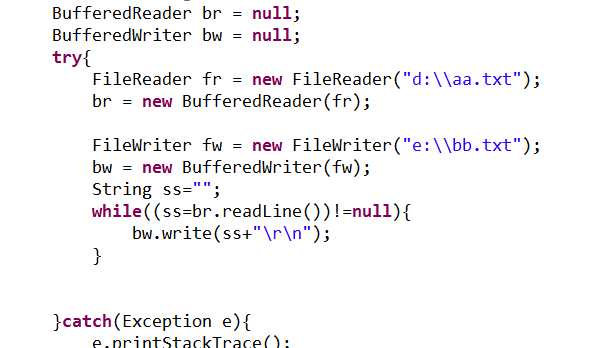
}

}

缓冲字符流，每次读取一行，再进行处理，可以提高效率

BufferedReader BufferedWriter

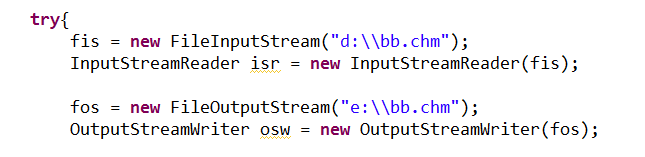
把fileReader和fileWriter包装一下形成br和bw



字节流可以转化为字符流通过InputStreamReader和OutputStreamWriter类

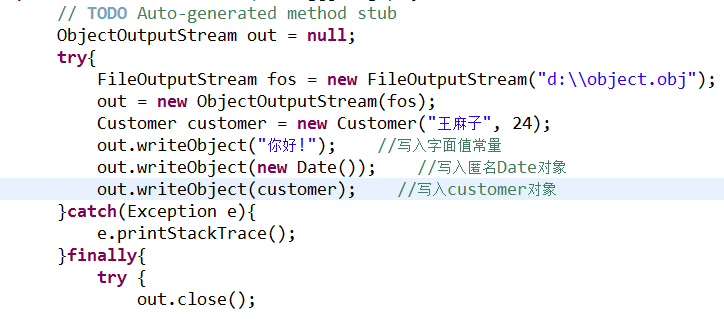
Eg1:

InputStreamReader isr = new InputStreamReader(System.in);

Eg2: 

对象序列化，需要实现Serializable接口

序列化对象，保存对象文件



反序列化对象，读取对象文件

