**IO**

1. **字节**

**注：字节批量读取:通过子节数组**

**1-1:RandomAccessFile**

Java 提供的对文件内容的访问，既可以读文件，也可以写文件

支持随机访问

可以访问文件的任意位置

Java文件模型

在硬盘上的文件是byte byte byte存储的，是数据的集合

打开文件

两种模式“rw”（读和写） “r” (只读)

RandomAccessFile raf=new RandomeAccessFile(file,”rw”);

文件指针，打开文件时，指针在开头 pointer=0;

写方法

raf.write(int)---🡪只写一个字节（后八位），同时指针指向下一个位置，准备再次写入

读方法

Int b=raf.read()---🡪读一个字节

文件读写完成以后一定要关闭（Oracle官方说明）

程序一：拷贝：

**public** **class** Kaobei {

**public** **static** **void** kaobei(File name2,File name1) **throws** IOException

{

RandomAccessFile f1=**new** RandomAccessFile(name1,"rw");

f1.write('1');

f1.write('A');

f1.write('B');

String s="中国";

**byte**[] b=s.getBytes("gbk");

f1.write(b);//将b中内容写入

f1.seek(0);

**int** y;

**byte** [] b2=**new** **byte**[(**int**)f1.length()];

f1.read(b2);//读入b2中

/\*\*

\* 复制

\*/

RandomAccessFile f2=**new** RandomAccessFile(name2,"rw");

f2.write(b2);

System.***out***.println(f2.equals(f1));

}

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

// **TODO** Auto-generated method stub

File f=**new** File("D:\\demo");

**if**(!(f.exists()))

{

f.mkdirs();//建立目录

}

File name1=**new** File(f,"a.txt");

**if**(!(name1.exists()))

{

name1.createNewFile();//建立文本

}

File name2=**new** File(f,"a1.txt");

**if**(!(name2.exists()))

{

name2.createNewFile();

}

Kaobei.*kaobei*(name2, name1);

}

}

程序二：

**public** **class** A {

/\*\*

\* 利用RandomAccessFile访问文件

\* **@param** name

\* **@throws** IOException

\*/

**public** **static** **void** c(File name) **throws** IOException

{

RandomAccessFile r=**new** RandomAccessFile(name,"rw");

/\*\*

\* 文件内容写入

\*/

r.write('1');

r.write('A');

r.write('B');

**int** i=12;

// r.write(i>>>24);

// r.write(i>>>16);

// r.write(i>>>8);

// r.write(i);

String s="中国";//字符串需要用字节数组装入，将字符串以字节数组的形式输入

**byte** []b=s.getBytes("gbk");//表明（输入字符串）文本格式

r.write(b);

System.***out***.println(r.getFilePointer());

System.***out***.println(r.length());

System.***out***.println("");

/\*\*

\* 读取文件内容

\*/

r.seek(0);//（RandomAccessFile）读文件之前要确定指针的位置

**int** d;

**while**((d=(r.read()))!=-1)

{

System.***out***.print(d);

}

System.***out***.println("");

/\*\*

\* 一次性读取

\*/

System.***out***.println("数组形式");

r.seek(8);

**byte**[] b1=**new** **byte**[(**int**)r.length()];//确定字节数组长度刚好为文件的大小

r.read(b1);//批量读取

System.***out***.print(Arrays.*toString*(b1));//以数组的形式输出

System.***out***.println("");

/\*\*

\* 一次性读取

\*/

System.***out***.println("16进制");

**for**(**byte** d1:b1)

{

System.***out***.print(Integer.*toHexString*((d1&0xff))+" ");

}

}

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

// **TODO** Auto-generated method stub

File name=**new** File("D:\\金宝\\jin.txt");

**if**(!(name.exists()))

{

name.createNewFile();

}

1. *c*(name);

*}*

}

**1-2文件输入流：FileInputStream**

IO流（输入流\输出流）

字节流、字符流

1. 字节流

(1)InputStream、OutputStream

InputStream 抽象了应用程序读取数据的方式

OutputStream 抽象了应用程序写出数据的方式

(2)EOF=end 读到-1就读到结尾

(3)输入流基本方式

Int b=in.read();读取一个字节无符号低八位

in.read(byte[] buf)一次将字节读入到数组中

in.read(byte[] buf,int start,int size)

(4)输出流基本方法

Out.write(int b) 写出一个byte到流，b的低八位

Out.write(byte[] buf) 一次写出

Out.write(byte[] buf,int start,int size)字节数组buf ， start位置开始写size长度的字节

**IO读取完之后进行关闭**

Eg:

**public** **class** A {

/\*\*

\* 类（InputStream）的具体实现类(读)

\* **@param** args

\* **@throws** IOException

\*/

**public** **static** **void** c(File name) **throws** IOException

{

FileInputStream s1=**new** FileInputStream(name);

RandomAccessFile r=**new** RandomAccessFile(name,"rw");

**byte** [] b=**new** **byte**[(**int**)r.length()];//不浪费开辟的空间

**int** i;

**int** j=1;

**while**((i=(s1.read()))!=-1)

{

**if**(i<=0xf)

System.***out***.print("0");

System.***out***.print(Integer.*toHexString*((i)&0xff)+" ");

**if**((j++)%10==0)

{

System.***out***.println();

}

}

}

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

// **TODO** Auto-generated method stub

File name=**new** File("D:\\金宝\\jin.txt");

A.*c*(name);

}

}

**1-3文件输出流：FileOutputStream**

**public** **class** A {

/\*\*

\* 类（Output）的具体实现类（写）

\* **@param** args

\* **@throws** IOException

\*/

**public** **static** **void** c(File name) **throws** IOException

{

**//如果该文件不存在则会直接创建，如果存在，删除后创建**

FileOutputStream f1=**new** FileOutputStream(name);**//name 后面如果有true则是在源文件后面补充**

f1.write('A');

f1.write('B');

f1.write('1');

String s="10";

**byte** [] b=s.getBytes("gbk");

f1.write(b);

String s1="中国";

**byte** [] b1=s1.getBytes("gbk");

f1.write(b1);

**//IOUtil.printHex(“文件”)；直接打印文件内容**

FileInputStream f2=**new** FileInputStream(name);

**int** i;

**int** j=1;

**while**((i=f2.read())!=-1)

{

**if**(i<=0xf)

System.***out***.print("0");

System.***out***.print(Integer.*toHexString*((i)&0xff)+" ");

**if**((j++)%10==0)

{

System.***out***.println();

}

}

}

**public** **static** **void** j (File name2,File name1) **throws** IOException

{

FileOutputStream f2=**new** FileOutputStream(name2);

FileInputStream f3=**new** FileInputStream(name1);

**int** i;

**while**((i=f3.read())!=-1)

{

f2.write(i);

}

}

**public** **static** **void** k(File name2,File name1) **throws** IOException

{

FileOutputStream f4=**new** FileOutputStream(name2);

FileInputStream f5=**new** FileInputStream(name1);

**byte** [] b5=**new** **byte**[(**int**)name1.length()];

**int** c;

**while**((c=f5.read(b5, 0, b5.length))!=-1)

{

f4.write(b5, 0, c);

f4.flush();

}

f4.close();

f5.close();

}

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

// **TODO** Auto-generated method stub

File name=**new** File("D:\\金玉波\\jyb.txt");

A.*c*(name);

File name1=**new** File("D:\\mp3\\周传雄 - 温柔的男人像海洋.mp3");

**if**(!(name1.exists()))

{

System.***out***.println("不存在此目录");

}

File name2=**new** File ("D:\\金玉波\\jinyubao.mp3");//复制的时候不需要用Creat创建

**long** start=System.*currentTimeMillis*();

//A.k(name2, name1);

A.*j*(name2, name1);

**long** end=System.*currentTimeMillis*();

System.***out***.println();

System.***out***.println(end-start);

}

}

**1-4数据输入输出流：DataOutputStream/ DataInputStream**

对“流”功能的扩展，可以更加方便的读取int，long,字符等类型数据

**DataOutputStream：**

WriteInt()/writeDouble()/writeUTF() (包装)

Eg:  
 **public** **class** A {

/\*\*

\* DataInputStream&DataOutputStream是对方法的封装类：eg;writeInt();

\* **@param** args

\* **@throws** IOException

\*/

**public** **static** **void** D(File name1) **throws** IOException

{

/\*\*

\* 写入文本

\*/

DataInputStream d1=**new** DataInputStream(**new** FileInputStream(name1));

DataOutputStream d2=**new** DataOutputStream(**new** FileOutputStream(name1));

d2.writeInt(13);

d2.writeInt(14);

d2.writeInt(179);

d2.writeUTF("中国梦");**//采用utf-8写出**

d2.writeUTF("圆梦");

d2.writeChars("我");**//采用utf-16be方式写出**

d2.close();**//写完记得关闭**

**//IOUtil.printHex(“文件”) 直接以16进制方式打印出来内容**

**/\*\***

**\* 读出来**

**\*/**

**int** i;

i=d1.readInt();

System.***out***.println(i);

i=d1.readInt();

System.***out***.println(i);

i=d1.readInt();

System.***out***.println();

String s;

s=d1.readUTF();

System.***out***.println(s);

s=d1.readUTF();

System.***out***.println(s);

d1.close();

d2.close();

}

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

// **TODO** Auto-generated method stub

File name1=**new** File("D:\\jyb1.txt");

// if(!(name1.exists()))

// {

// name1.createNewFile();

// }

A.*D*(name1);

}

}

**1-5字节缓冲流：BufferedInputStream& BufferedOutputStream**

这两个流类为IO提供了带缓冲区的操作，一般打开文件进行写入或读取操作时，都会加上缓冲，**这种流模式提高了IO的性能（注意加上（.flush））**

**FileOutputStream----》write()方法：相当于一滴一滴的把水移过去**

**DataOutputStream----》write()方法：相当于一瓢一瓢把水移过去**

**BufferedOutputStream—》write()方法：相当于一瓢一瓢先放入桶中，再把水移过去缸中**

**public** **class** A {

**public** **static** **void** C(File name,File name1) **throws** IOException

{

BufferedInputStream b=**new** BufferedInputStream(**new** FileInputStream(name));

BufferedOutputStream b1=**new** BufferedOutputStream(**new** FileOutputStream(name1));

**byte** b2[]=**new** **byte**[(**int**)name.length()];

b.read(b2, 0, b2.length);

b1.write(b2, 0, b2.length);

b1.flush();//缓冲池必须要弄这个

b1.close();

b.close();

// int c;

// while((c=b.read())!=-1)

// {

// b1.write(c);

// b1.flush();//刷新

// }

}

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

// **TODO** Auto-generated method stub

File name=**new** File("D:\\美文\\a.txt");

File name1=**new** File("D:\\美文\\a3.txt");

**if**(!(name1.exists()))

{

name1.createNewFile();

}

**long** start=System.*currentTimeMillis*();

A.*C*(name, name1);

**long** end=System.*currentTimeMillis*();

System.***out***.println(end-start);

}

}

**//格式化一下这段代码 右键：如图**

1. 字符流------》操作的是文本文件

2-1: 编码问题

认识文本和文本文件

Java的文本（char）是16位无符号整数，是字符的unicode编码（双字节编码）

文本文件是文本（char）序列按照某种编码方式（utf-8，utf-16be,gbk）序列化为byte的存储结果

字符流（Reader Writer）

字符的处理，一次处理一个字符

字符的底层仍然是最基本的字节序列

字符流的基本实现

InputStreamReader 完成byte流解析为char流，按照编码解析

OutputStreamWriter 完成char流解析为byte流，按照编码解析

Eg;

**public** **class** A {

**public** **static** **void** j(File name1,File name) **throws** IOException

{

InputStreamReader i=**new** InputStreamReader (**new** FileInputStream(name),"gbk");

**//注意后面有个参数指明编码方式,默认按照gbk方式解析的，此java项目是按照gbk方式编码的**

OutputStreamWriter o=**new** OutputStreamWriter(**new** FileOutputStream (name1));

**int** c;

/\*\*

\* 读取文件内容

// \*/

// while((c=i.read())!=-1)

// {

// System.out.println((char)c);

// }

// System.out.println();

/\*\*

\* 批量读取

\*/

**int** h;

**char** [] b=**new** **char**[290\*1024];

**while**((h=i.read(b, 0, b.length))!=-1)

{

**String s=new String(b, 0, h);**

**System.*out*.print(s);//char字符组转化为字符串**

o.write(b,0,h);

o.close();

i.close();

}

System.***out***.println();

/\*\*

\* 复制文件

\*/

**int** h1;

// while((h1=i.read())!=-1)

// {

// o.write(h1);

// o.flush();

// }

o.close();

i.close();

}

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

// **TODO** Auto-generated method stub

File name=**new** File("D:\\美文\\a3.txt");

File name1=**new** File("D:\\fuzhi.txt");

**if**(!(name1.exists()))

{

name1.createNewFile();

}

A.*j*(name1, name);

}

}

2-2：FileWiter&FileReader

可以直接对文件进行读写

Eg:  
**public** **class** A {

**public** **static** **void** j(File name1,File name) **throws** IOException

{

FileReader f1=**new** FileReader(name1);**//默认解析项目默认的编码对应的文件类型**

FileWriter f2=**new** FileWriter(name);**//追加，后面的参数加true,参数里面没有编码的参数，回归到2-1**

**char** [] c=**new** **char**[200\*1024];

**int** c1;

/\*\*

\* 批量读取复制

\*/

**while**((c1=f1.read(c, 0, c.length))!=-1)

{

String s=**new** String(c, 0, c1);

System.***out***.print(s);

// f2.write(c, 0, c1);

// f2.flush();

}

f1.close();

f2.close();

}

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

// **TODO** Auto-generated method stub

File name=**new** File("D:\\美文\\fuzhi.txt");

File name1=**new** File("D:\\fuzhi1.txt");

**if**(!(name1.exists()))

{

name1.createNewFile();

}

A.*j*(name1, name);

}

}