Chapter 9

Java I/O

In java, Byte stream based I/O it handle by InputStream, OutputStream.

Other, Character based I/O it handle by Reader, Writer class.

Stream

Byte Stream Character Stream

InputStream OutputStream Reader Writer

Path:-

1. Relative Path:- “test.txt”
2. Absolute Path:- “c:\user\roy\test.txt”

Create file:-

Constructor of File Class:

1. File(String pathname)
2. File(File parent, String child)
3. File(String parent, String child)
4. File(URL url)

Method :-

isFile() & isDirectory() to check it is directory or file.

File f=new File(“test.txt”);

If(!f.exists)

f.createNewFile();

System.out.println(f.isFile());

// current directory

String dir=System.getProperty(“user.dir”);

Path Separator:-

String newFile=dir+File.separator+”test.txt”;

Create Directory:-

We can create directory using mkdir() & mkdirs() method.

File dir=new File(“home/user/newDir”);

Dir.mkdir(0;

String path=dir.getPath();

File Renaming:-

We can rename file using renameTo() method.

File oldFile=new File(“old.txt”);

File newFile=new File(“new.txt”);

Boolean fileRename=oldFile.renameTo(newFile);

Delete File:-

We can delete file using delete() & deleteOnExit() method.

delete() method is delete file .

deleteOnExit() method delete file when JVM close.

Get File List:-

We can get list of files using listFiles() method.

File home=new File(“/home/”);

File[] fileList=home.listFile();

Filter File:-

When we filter just ‘.png’ file.

File home=new File(“/home/picture”);

FileFilter pngFilter=new FileFilter()

{

@Override

Public Boolean accept(File pathName)

{

String filename=pathname.getName();

If( filename.endWith(“.png”)

Return true;

Return false;

}

};

File[] listpics=home.listFiles(pngFilter);

Input Output Stream:-

Every Stream have Source & Destination. Source called Data Source & Destination called Data Sink.

Read Data from Input Stream:-

Java.io.InputStream is an Abstract class & Super class of every Input Stream Class.

Byte Stream:

FileInputStream in=null;

in=new FileInputStream(“input.txt”);

int c;

while((c=in.read()!=-1) // return 1 byte

{

System.out.println(c+” “+(char) c);

}

in.close();

Output Stream:-

Make Byte:

String text=”Hello”;

byte[] textbyte=text.getByte();

Ex:-

String destFile=”output.txt”;

String data=”My name is Pollob”;

FileOutputStream fos=new FileOutputStream(destFile);

fos.write(data.getBytes());

fos.flush(); // it will clear the Buffer

fos.close();

Character Stream:-

Read data using Reader class:

Reader reader=null;

Reader=new FileReader(“input.txt”);

Int c;

While((c==reader.read())!=-1) // return 1 character

{

Char ch=(char) c;

System.out.println(ch);

}

Writer:-

Writer writer=new FileWriter(“output.txt”);

String text=”my name is pollob”;

writer.write(text);

writer.flush();

writer.close();

System.in System.out System.error

System.in:

System.in is a Input Stream, এটি কনসোল এ ভিত্রিক প্রোগ্রাম এর জন্য কিবোর্ড এর সাথে সংযুক্ত।

Ex:

Scanner sc=new Scanner(System.in);

System.out:

System.out হল একটি OutputStream. এটি Console এ write করে। এটি একটি Output Stream। এটি Data Format করে দেখতে সাহায্য করে।

System.error:

System.error একটি Output Stream যা System.out Stream এর মত কাজ করে। তবে এটি শুধুমাত্র Error Print করে।

Read Primitive Datatype:-

DataInputStream & DataOutputStream class has two method for primitive data read/write, readXxx() & writeXxx().

DataInputStream dis=new dataInputStream(new FileInputStream(“primitive.data”));

int intval=dis.readInt();

double dval=dis.readDouble();

boolean bval=dis.readBoolean();

Write:-

DataOutputStream dos=new DataOutputStream(new FileOutputStream(“Primitive.data”));

dos.writeInt(152);

dos.writeDouble(4.56);

Convert data into Upper case:

BufferedReader reader=new bufferReader(new InputStreamReader(System.in));

String line;

do{

line=reader.readLine();

line.toUpperCase();

System.out.println(line);

}

While(!”QUIT”.equals(line));