IO包中绝大部分的类都是由以下四个类直接或间接继承来的 InputStream OutputStream Reader 还有Writer

其中InputStream和OutputStream代表输入流和输出流,也就是字节流的输入和输出,他们定义了如何读取和写入字节和字节数组,所以说基本上所有XXXInputStream和XXXOutputStream都是针对字节进行操作的比如说FileInputStream,它可以以流的形式读取一个文件,或者StringBufferInputStream,它以流的形式读取一个字符串,所有的子类都是不同领域的应用罢了。

而Reader和Writer是在输入输出流之上的更高级的字符级别的输入输出,称为读取器和写入器,他们直接读取和写入字符(字符串)数据而不是字节(字节数组),比如你有一个文本文件就可以使用FileReader这个类来读取里面的文本,还有PrintWriter是用来输出的写入器,System.out的那个out返回的就是一个PrintWirter的内部实现PrintOutputStream

一个例子完全搞定。

package com.sinovatech.fmcc.ehall2.common;

/\*\*

\* @version 1.0

\* @author

\*

\* 多种方式读文件的内容

\* 按字节读取文件内容,按字符读取文件的内容,按行读取文件的内容,随即读取文件的内容

\*/

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.RandomAccessFile;

import java.io.Reader;

import java.util.ResourceBundle;

public class ReadFromFile {

/\*\*

\* 以字节为单位读取文件的内容,常用于二进制文件,如声音,图象,影象等文件

\*

\* @param filename

\* 文件名

\*/

public static void readFileByBytes(java.lang.String filename) {

File file = new File(filename);

InputStream in = null;

System.out.println("以字节为单位读取文件的内容,一次读一个字节: ");

// 一次读一个字节

try {

in = new FileInputStream(file);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

int tempbyte;

try {

// 不断的读取,直到文件结束

while ((tempbyte = in.read()) != -1) {

System.out.write(tempbyte);

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

in.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

return;

}

System.out.println("以字节为单位读取文件内容,一次读多个字节: ");

// 一次读取多个字节

byte[] tempbytes = new byte[100];

int byteread = 0;

try {

in = new FileInputStream(filename);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

ReadFromFile.showAvailabelBytes(in);

try {

while ((byteread = in.read(tempbytes)) != -1) {

// 读取多个字节到数组中,byteead为一次读取的字节数

System.out.write(tempbytes, 0, byteread);

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void readFileByteChars(java.lang.String filename) {

/\*\*

\* 以字符为单位读取文件,常用与读文本,数字等类型的文件

\*/

File file = new File(filename);

Reader reader = null;

System.out.println("以字符为单位读取文件内容,一次读一个字节: ");

// 一次读一个字符

try {

reader = new InputStreamReader(new FileInputStream(file));

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

int tempchar;

try {

while ((tempchar = reader.read()) != -1) {

// 在Window下,\r\n这两个字符在一起时,表示一个换行

// 但如果这两个字符分开显示时,会换行两次行

// 因此,屏蔽掉\r,或者\n;否则,将会多出来很多空行

if (((char) tempchar) != '\r') {

System.out.println((char) tempchar);

}

}

reader.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("以字符为单位读取文件内容,一次读多个字符: ");

char[] tempchars = new char[30];

int charread = 0;

try {

reader = new InputStreamReader(new FileInputStream(filename));

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

// 读入多个字符到字符数组中,charread为一次读取字符数

while ((charread = reader.read(tempchars)) != -1) {

if ((charread == tempchars.length)

&& (tempchars[tempchars.length - 1] != '\r')) {

System.out.println(tempchars);

} else {

for (int i = 0; i < charread; i++) {

if (tempchars[i] == '\r') {

continue;

} else {

System.out.print(tempchars[i]);

}

}

}

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

if (reader != null) {

try {

reader.close();

} catch (IOException e) {

}

}

}

}

public static void readFileByLines(java.lang.String filename) {

File file = new File(filename);

BufferedReader reader = null;

// System.out.println("以行为单位读取文件的内容,一次读一整行: ");

try {

reader = new BufferedReader(new FileReader(file));

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

java.lang.String tempString = null;

int line = 1;

try {

while ((tempString = reader.readLine()) != null) {

if(tempString.indexOf("福州")!=-1){

System.out.println("出错了！赶紧报警啊！ " + line + ": " + tempString);

}

// System.out.println("line " + line + ": " + tempString);

line++;

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

reader.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

if (reader != null) {

try {

reader.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

public static void readFileByRandomAccess(java.lang.String filename) {

RandomAccessFile randomFile = null;

System.out.println("随即读取一段文件内容: ");

try {

randomFile = new RandomAccessFile(filename, "r");

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

long fileLength = 0;

try {

fileLength = randomFile.length();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

int beginIndex = (fileLength > 4) ? 4 : 0;

try {

randomFile.seek(beginIndex);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

byte[] bytes = new byte[10];

int byteread = 0;

try {

while ((byteread = randomFile.read(bytes)) != -1) {

System.out.write(bytes, 0, byteread);

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

if (randomFile != null) {

try {

randomFile.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

private static void showAvailabelBytes(InputStream in) {

try {

System.out.println("当前字节输入流中的字节数为: " + in.available());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void main(java.lang.String args[]) {

ResourceBundle rb = ResourceBundle.getBundle("PreventTampering");

String filePath = rb.getString("filePath");

java.lang.String filename = filePath+"test.txt";

System.out.println(filename);

// ReadFromFile.readFileByBytes(filename);

// ReadFromFile.readFileByteChars(filename);

ReadFromFile.readFileByLines(filename);

// ReadFromFile.readFileByRandomAccess(filename);

}

}