1. FileReader / FileWriter的使用：
   1. FileReader的使用

*/\*  
将day09下的hello.txt文件内容读入程序中，并输出到控制台  
说明点：  
1. read()的理解：返回读入的一个字符。如果达到文件末尾，返回-1  
2. 异常的处理：为了保证流资源一定可以执行关闭操作。需要使用try-catch-finally处理  
3. 读入的文件一定要存在，否则就会报FileNotFoundException。  
 \*/*

@Test  
 public void testFileReader1() {  
 FileReader fr = null;  
 try {  
 *//1.File类的实例化* File file = new File("hello.txt");  
 *//2.FileReader流的实例化* fr = new FileReader(file);  
 *//3.读入的操作  
 //read(char[] cbuf):返回每次读入cbuf数组中的字符的个数。如果达到文件末尾，返回-1* char[] cbuf = new char[5];  
 int len;  
 while((len = fr.read(cbuf)) != -1){ *//每次讀取完都刷新len的長度*  
 *//方式一：  
 //错误的写法  
// for(int i = 0;i < cbuf.length;i++){  
// System.out.print(cbuf[i]);//helloworld123ld  
// }  
 //正确的写法  
// for(int i = 0;i < len;i++){  
// System.out.print(cbuf[i]);//helloworld123  
// }  
 //方式二：  
 //错误的写法,对应着方式一的错误的写法  
// String str = new String(cbuf);  
// System.out.print(str);//helloworld123ld,*

*因為如果cbuf最後一次沒裝滿,上一次剩下沒被刷新到的(ld)會再印出來  
 //正确的写法* String str = new String(cbuf,0,len);  
 System.*out*.print(str);*//helloworld123* }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 if(fr != null){  
 *//4.资源的关闭* try {  
 fr.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }

* 1. FileWriter的使用

*/\*  
从内存中写出数据到硬盘的文件里。  
说明：  
1. 输出操作，对应的File可以不存在的。并不会报异常，系統會自動幫你造一個  
2.  
 File对应的硬盘中的文件如果不存在，在输出的过程中，会自动创建此文件。  
 File对应的硬盘中的文件如果存在：  
 如果流使用的构造器是：FileWriter(file,false) / FileWriter(file):对原有文件的覆盖  
 如果流使用的构造器是：FileWriter(file,true):不会对原有文件覆盖，而是在原有文件基础上追加内容  
 \*/*

@Test  
public void testFileWriter() {  
 FileWriter fw = null;  
 try {  
 *//1.提供File类的对象，指明写出到的文件* File file = new File("hello1.txt");  
 *//2.提供FileWriter的对象，用于数据的写出* fw = new FileWriter(file,false);  
 *//3.写出的操作* fw.write("I have a dream!\n");  
 fw.write("you need to have a dream!");  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 *//4.流资源的关闭* if(fw != null){  
 try {  
 fw.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

* 1. 文本文件的複製

@Test  
 public void testFileReaderFileWriter() {  
 FileReader fr = null;  
 FileWriter fw = null;  
 try {  
 *//1.创建File类的对象，指明读入和写出的文件* File srcFile = new File("hello.txt");  
 File destFile = new File("hello2.txt");  
 *//不能使用字符流来处理图片等字节数据  
// File srcFile = new File("爱情与友情.jpg");  
// File destFile = new File("爱情与友情1.jpg");  
 //2.创建输入流和输出流的对象* fr = new FileReader(srcFile);  
 fw = new FileWriter(destFile);  
 *//3.数据的读入和写出操作* char[] cbuf = new char[5];  
 int len;*//记录每次读入到cbuf数组中的字符的个数* while((len = fr.read(cbuf)) != -1){  
 *//每次写出len个字符* fw.write(cbuf,0,len);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 *//4.关闭流资源  
 //方式一：  
// try {  
// if(fw != null)  
// fw.close();  
// } catch (IOException e) {  
// e.printStackTrace();  
// }finally{  
// try {  
// if(fr != null)  
// fr.close();  
// } catch (IOException e) {  
// e.printStackTrace();  
// }  
// }  
 //方式二：因為異常已經被try\_catch處理掉了,所以下面的程式一定會去執行* try {  
 if(fw != null)  
 fw.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 try {  
 if(fr != null)  
 fr.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }

1. FileInputStream / FileOutputStream的使用：
2. *对于文本文件(.txt,.java,.c,.cpp)，使用字符流处理*

*2. 对于非文本文件(.jpg,.mp3,.mp4,.avi,.doc,.ppt,...)，使用字节流处理*

*/\*  
实现对图片的复制操作  
 \*/*@Test  
public void testFileInputOutputStream() {  
 FileInputStream fis = null;  
 FileOutputStream fos = null;  
 try {  
 *//1.造文件* File srcFile = new File("爱情与友情.jpg");  
 File destFile = new File("爱情与友情2.jpg");  
 *//2.造流* fis = new FileInputStream(srcFile);  
 fos = new FileOutputStream(destFile);  
 *//3.复制的过程* byte[] buffer = new byte[5];  
 int len;  
 while((len = fis.read(buffer)) != -1){  
 fos.write(buffer,0,len);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 if(fos != null){

*//4.關閉流*  
 try {  
 fos.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if(fis != null){  
 try {  
 fis.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

[注意]

相對路徑在IDEA和Eclipse中使用的區別?

IDEA：

如果使用單元測試方法，相對路徑基於當前的Module的

如果使用main()測試，相對路徑基於當前Project的

Eclipse：

單元測試方法和main()，相對路徑都是基於當前Project的