复习：

1. 迭代器：Iterator - 通过状态标记和指针的挪动来遍历元素 - 在迭代过程中不允许直接增删原集合。Iterable接口提供了获取迭代器的方法，实现了这个接口的类所产生的对象可以被增强for循环遍历

2. 映射：用于存储键值对的容器。每一个键对应一个值，键是唯一的。

HashMap：底层是基于哈希码进行存储，使用的是数组+链表结构。默认初始容量是16，默认加载因子是0.75f，每次默认增加一倍。容量是一定是2n的形式。允许键或者值为null，本身是一个异步式线程不安全的映射

Hashtable：底层也是基于哈希码进行存储，使用的是数组+链表结构。默认初始容量是11，默认加载因子是0.75f，每次默认增加一倍，再+1。不允许键和值为null。本身是一个同步式线程安全的映射。

3. 断言：用于预测结果。断言不是默认开启，需要添加参数-ea来手动开启断言。assert 预测条件 : 提示信息;

## File

代表文件或者目录（文件夹）的类

练习：删除目录

思路：写一个单独的方法来删除目录或者文件：判断是一个目录还是还是一个文件；如果是文件(isFile())直接删除(delete())，如果是目录(isDirectory())，获取这个目录下的所有的子目录和子文件(listFiles()) --- 后续功能和当前方法要完成的事情是一致的，所以使用递归

练习：统计工作空间中Java文件和class文件的个数

思路：定义一个新的方法进行统计：先获取工作空间中所有的子文件和子目录。如果是子目录---递归；如果子文件，需要判断这个文件是否是一个Java文件还是一个class文件

### 路径

绝对路径：以盘符或者/开头的路径。指定的位置和文件当前错处的位置没有任何关系

相对路径：不以盘符或者/开头的路径。以当前路径为基准来计算指定的文件的所在路径

## IO流

用于数据的传输的机制。IO -> Input Output Stream -> 输入输出流。数据从外部流向程序 - 输入流； 数据从程序流向外部 - 输出流。读取文件 -> 数据是从文件读到程序中 -> 输入流；向一个TXT文件中写入字符串 “abc” -> 数据是从程序写到文件中 -> 输出流

根据数据传输方向：输入流和输出流

根据数据传输形式：字符流和字节流

|  |  |  |
| --- | --- | --- |
|  | 输入流 | 输出流 |
| 字符流 | Reader | Writer |
| 字节流 | InputStream | OutputStream |

四个基本的流都是抽象类。

数据的来源/目的地：存储设备、物理设备、内存、网络

向一个TXT文件中写入字符串 -> 输出流、字符流、和文件相关的流 -> FileWriter

### 流中的异常处理

1. 将流对象放在try之外声明并且赋值为null；放到try之内进行初始化

2. 在关流之前需要判断流对象是否为空

3. 为了防止关流失败导致流依然占用文件，所以需要将流对象强制置为null

4. 为了防止关流的时候自动冲刷缓冲区失败导致一部分数据产生丢失，需要在关流之前进行一次手动的冲刷