第一组：

问题一：如果在一个接口中将一个方法定义为默认方法，在父类或者另一个接口中定义同样的方法会调用哪一个方法？哪一个调用的优先级更高？

➔ https://blog.51cto.com/jamesdev/1857186

问题二：List、Set、Map 三种数据结构在平时程序中的具体应用场景及使用时的注意点和区别？

➔ List 元素有放入顺序，元素可重复，和数组类似，List 可以动态增长，查找元素效率高，插入删除元素效率低，因为会引起其他元素位置改变。

Set 元素无放入顺序，元素不可重复，重复元素会覆盖掉，Set 检索元素效率低下，删除和插入效率高，插入和删除不会引起元素位置改变。

Map 是一个接口，适合储存键值对的数据

使用场景：

1. 如果你经常会使用索引来对容器中的元素进行访问，那么 List 是你的正确的选择
2. 如果你想容器中的元素能够按照它们插入的次序进行有序存储，那么还是 List，因为 List 是一个有序容器，它按照插入顺序进行存储
3. 如果你想保证插入元素的唯一性，也就是你不想有重复值的出现，那么可以选择一个 Set 的实现类
4. 如果你以键和值的形式进行数据存储那么 Map 是你正确的选择

原文链接：https://blog.csdn.net/johnWcheung/article/details/79889882

具体讨论 1：有关于读写文件程序的具体写法和所遇到的问题

题问 1：具体最后文件程序的读写如何实现？

回答：例程中有分别对应于文件建立，读文件，写文件的代码块，把不同的功能相整合即可。

追问 1: 为什么我这里的关于输入的声明会报错？

回答：你要对具体的文件路径进行设置，例程里是提供了固定的测试文件，因此是 privatestaticfinalstring 的格式去定义的，你可以通过 Scanner 去采当前路径下你想要编辑的文件名，然后通过 file 类文件的格式和引用把具体的对应文件路经设置好，你按例程那样写只能对你规定的测试文件进行调试。

讨论小结 1：异常对象的来源和各种分类：

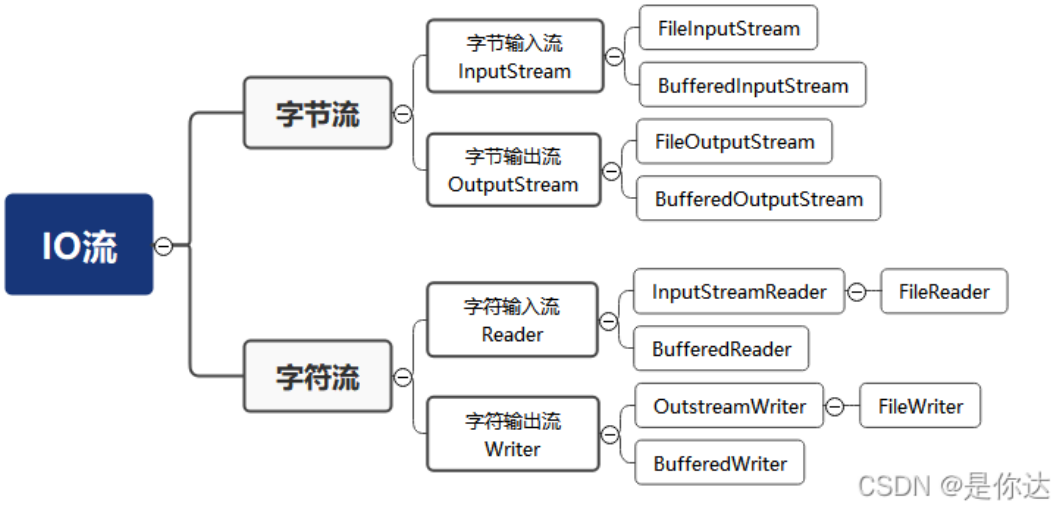
在 Java 中，异常对象都是派生于 Throwable 类的类实例，除了 Java 内置的异常类以外，若不满足需求我们还可以创建自己的异常类。所有的异常可以分为 Error 和 Exception 两个大类，其中 Error 主要是指 Java 运行过程中系统的内部错误和资源耗尽问题。这种问题往往是不应该被抛出的，因为除了终止程序难以对其做出影响。而 Exception 又可以分为 IOException 和 RuntimeException 两种，由编译产生的错误一般源自于 RuntimeException，而像 I/O 错误等问题一般属于其他异常。

讨论小结 2：List，Set 以及 Map

数组是在连续的存储位置上存放对象引用，而 List 链表的主要特征是将每个对象存放在单独的 Link 中，每个链接存放着序列中下一个链接的引用，很方便在一个链表中插入或者去除一个元素。List 和 Set 的主要区别在于 List 是一个有序的集合，每个对象的位置是固定且重要的，而 Set 类型中，元素是完全无序的，而 Set 的主要特点是没有重复的元素的元素集合，为 Set 添加元素的一般方法是先在这个集中查找这个对象，若不存在该元素即添加。Map，映射数据结构是为了通过某些关键的信息来方便查找与之相对的元素，和 Set 和 List 这种动厩遍历全体元素的查找方法不同，Map 主要是用来存放 key/value 对，若提供了 key 应该就能查找到对应的值。

第二组：

1. JAVA 写文件的几种方法



链接：

https://blog.csdn.net/weixin_44047784/article/details/122130834?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-0-122130834-blog-118446582.pc_relevant_paycolumn_v3&spm=1001.2101.3001.4242.1&utm_relevant_index=3

2. 两种方式的对比

法 1：

```
PrintWriter pw = new PrintWriter(file);
```

法 2：

```
FileOutputStream fos = new FileOutputStream(targetFile,append);
OutputStreamWriter osw = new OutputStreamWriter(fos, StandardCharsets.UTF_8);
PrintWriter pw = new PrintWriter(osw);
```

PrintWriter 在 PrintWriter.java 中有多种方法定义。
法 1 中为向文件写入，这种简化的方式下，JDK 将 OutputStreamWriter 给创建好了。且文件写入自动覆盖旧内容。
法 2 通过调整 append 状态可定义写入是覆盖或添加。

3. 内部类的作用？成员内部类、静态内部类、局部内部类区别？

作用：

1、 内部类方法可以访问该类定义所在的作用域中的数据，**包括私有数据。**
2、 内部类可以对同一个包中的其他类隐藏起来。

静态内部类

1.静态内部类可以等同的看作是静态变量
内部类的重要作用：可以访问外部类中的私有数据
2.静态内部类可以直接访问外部类的静态变量，静态方法，无法直接访问成员变量和成员方法。

成员内部类

1.成员内部类可以等同的看作成员变量。
2.成员内部类中不允许存在静态变量、静态方法、只能有成员变量和成员方法。
3.成员内部类可以有外部类的成员变量、成员方法、静态变量和静态方法。

局部内部类

1、不能有 private、public、protected 和 static 等修饰符，与局部变量类似。
2、只能在定义局部内部类的方法或者作用域中实例化；
3、局部内部类的对象不能使用该内部类所在方法或者作用域的非 final 局部变量

链接：

https://blog.csdn.net/sulixu/article/details/120028541
https://blog.csdn.net/awodwde/article/details/109497755

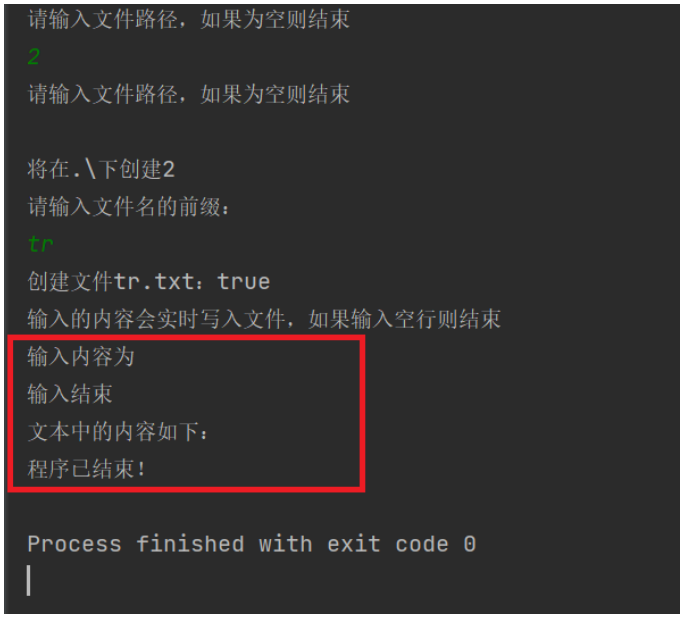
第三组：

问题：JAVA 除了抛出异常，是否还有其他的异常提示语法吗？比如：我要从键盘读入一个整数，范围是 0-10，当用户输入的数超出这个范围的话，在控制台输出自定义的警告信息或者错误消息。

➔ **这个取决于项目的需求，可以抛异常，也可以打印警告和错误消息。**

第四组：

问题：在写入和读取小程序中，输入并创建好文件路径和名称后，程序写入和读取所需的 scanner 类中的 nextLine()方法没有执行，而是直接执行后面的程序。



解决办法:在写入文本时再次声明 Scanner 类的对象。

原因：由于前后使用的 next（）方法不统一导致，使用 nextline()时，捕获到之前 next()遗留的空格。

其他解决办法：

1、统一 next（）或者 nextline（）方法

2、如果不想统一，nextLine()之前添加一个 nextLine()用于接收’\n’,这样就可以输入自己想要输入的字符串了。

知识点分享：

1、异常的使用原则：

Java 的异常是为了强制用户考虑程序的健全性和安全性，主要作用是捕获程序在运行时发生的异常并进行相应的处理，其不应该用来控制程序的正常流程，可遵循原则如下：

（1） 在当前方法声明中使用 try-catch 语句捕获异常。

（2） 一个方法被覆盖时，覆盖它的方法必须抛出相同的异常或异常的子类

（3） 如果父类抛出多个异常，则覆盖方法必须抛出那些异常的一个子类，不能抛出新异常。

注意：try-catch 语句中，在 catch 中写入处理异常的代码是关键，否则 try-catch 成了摆设。

2. List,Set,Map 用法以及区别

Map、list、set 三种都是集合用来存储对象类型的数据，其中 list 和 set 属于是属于单列集合，map 属于双列集合，

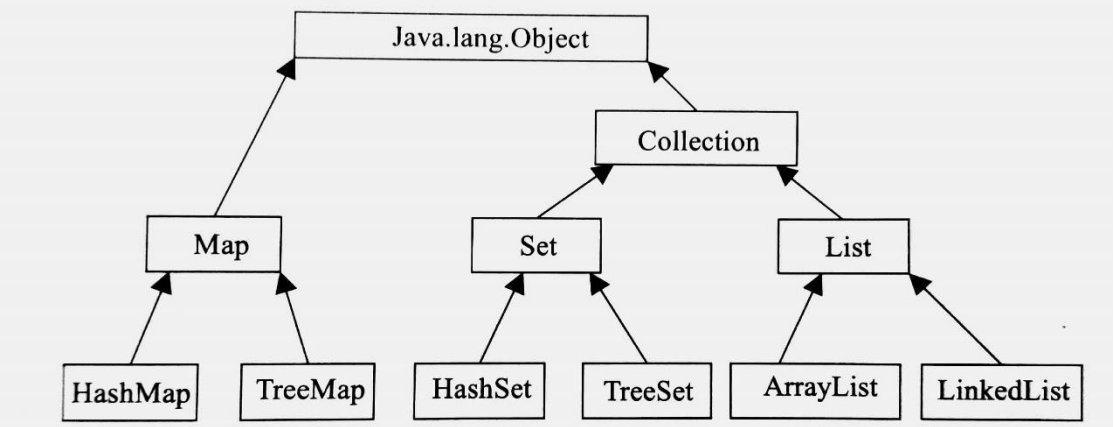


图 12.1　常用集合类的继承关系

list 集合的特点是元素有序且可重复，set 是元素无序不能重复，Map 集合以 Key 值对来存储元素。当需要元素唯一时，使用 set，否则使用 list，当需要 key 值对特性时，使用 Map。

与数组相同，集合的索引也是从 0 开始。

3. 常见的表示路径的符号

包含盘符的路径名前缀由驱动器号和一个“：”组成，如果是绝对路径名，后面还会有“\\”,如 D:\\test.txt 表示 D 盘下的 test 文本文件

“.”表示当前目录

“..”表示当前目录的上一级目录。

“./”表示当前目录下的某个文件或文件夹，视后面跟着的名字而定

“../”表示当前目录上一级目录的文件或文件夹，视后面跟着的名字而定。

“~”表示根目录