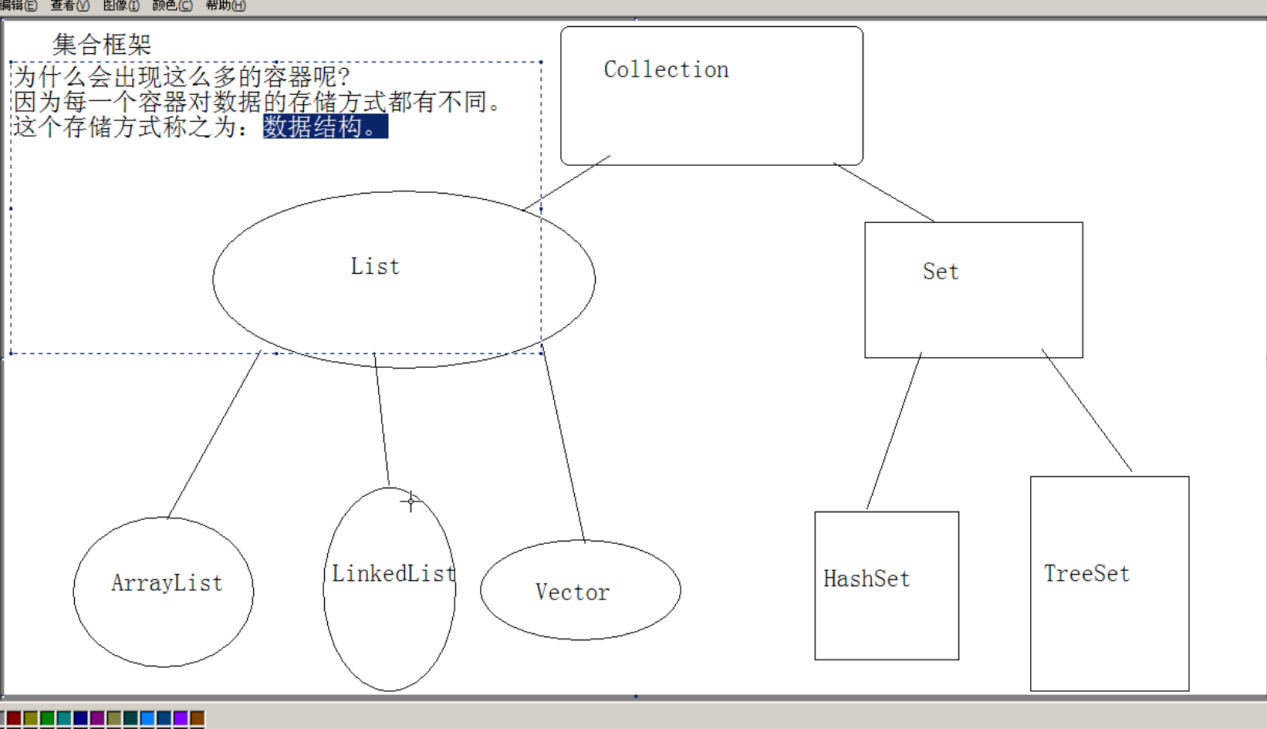
**Java学习day14**

1. **集合框架(体系概述) 对应166讲**

集合类是用来存储对象的，并且它的长度是可变的，集合可以存储不同的数据类型。



为什么会出现容器？

因为每一个容器对数据的存储方式都有不同，这个存储方式成为数据结构。

**2.集合框架(共性方法) 对应167讲** 可以自己通过查API完成

字符串也是一个对象。

集合类中存放的不是对象，而是对象对应的地址或者叫引用。

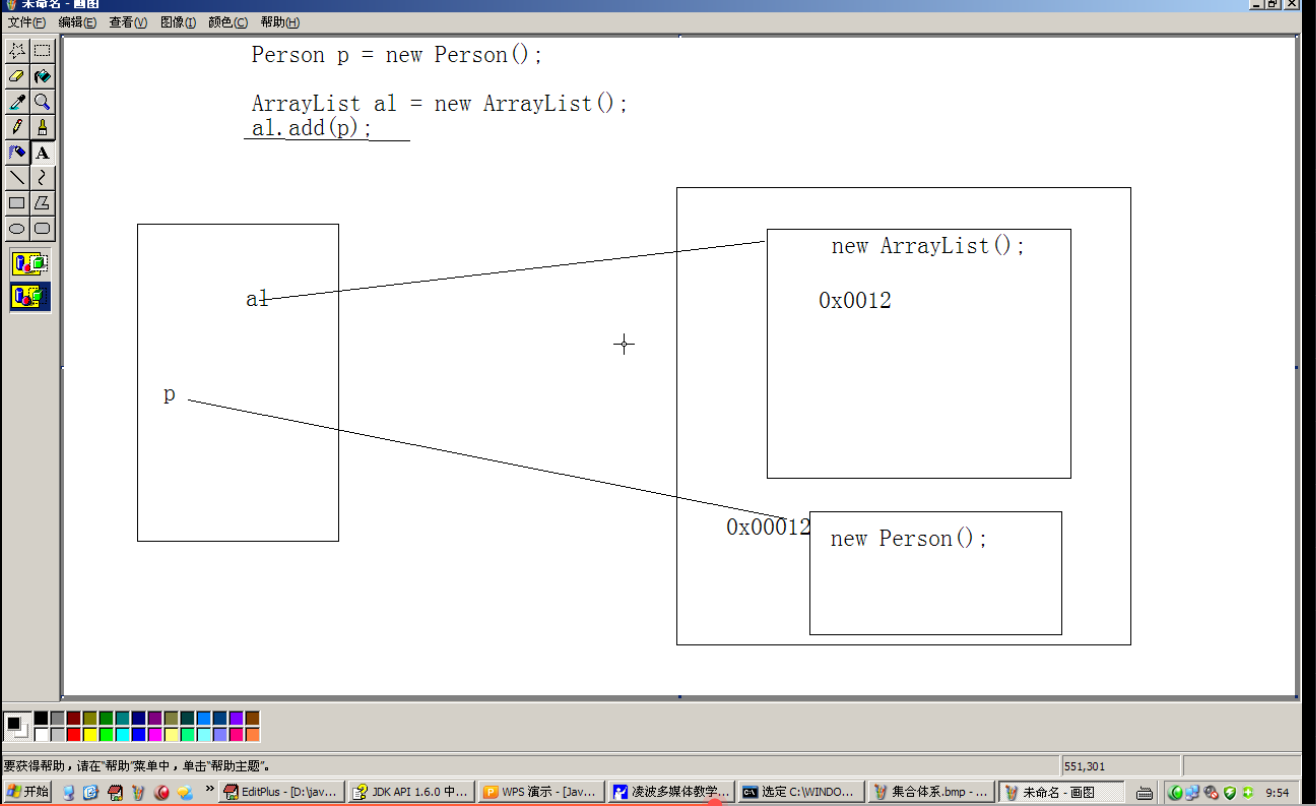
集合框架(共性方法)的流程：

1.导包。

2.创建一个集合容器，使用collection接口的子类，ArrayList

3.在这里做一下添加、获取及和长度，删除，判断，取交集的步骤

add方法接收的数据类型是Object，以便接收任意类型的数据对象。



**3.集合框架(迭代器)**

集合框架(共性方法)的流程：

1.导包。

2.创建一个集合容器，使用collection接口的子类，ArrayList

3.在这里做一下添加、获取及和长度，删除，判断，取交集的步骤

集合中存放的不是对象的实体，而是地址(或者说是引用)。

add方法接收的数据类型是Object，以便接收任意类型的数据对象。

用迭代器获取集合中的元素。 Iterator 自己查一下API手册了解

Iterator可以认为是一个取出娃娃机的夹子。

不同容器虽然数据结构不同，但是都有共同的取出方式，这里将这个取出方式定义为一个内部类，也就是Iterator类，通过操作它可以从不同的容器中取出元素。

记住其中的两个重要的方法 hasNext()和next()。

[hasNext](mk:@MSITStore:D:\\Learning%20materials\\java\\jdk1.8中文百度.CHM::/java/util/../../java/util/Iterator.html" \l "hasNext--)()

返回true如果迭代具有更多的元素。

[next](mk:@MSITStore:D:\\Learning%20materials\\java\\jdk1.8中文百度.CHM::/java/util/../../java/util/Iterator.html" \l "next--)()

返回迭代中的下一个元素。

1. **集合框架（ListIterator）**

集合框架(List集合共性方法)

Collection

|--List: 元素是有序的，元素可以重复，因为该集合有索引

|--ArrayList 底层数据结构是数组结构。特点：查询速度较快，增删速度较慢。线程不同步

|--LinkList 底层数据结构是链表数据结构。 特点：查询速度较慢，增删速度较快。

|--Vector 底层数据结构是数组数据结构。 特点：什么都很慢。 线程同步

|--Set：元素是无序的，元素不可以重复。List：

特有方法：凡是可以操作角标的方法都是该列表的特有方法。

增

add(index,element)

add(index,Collection)

删

remove(index)

改

set(index,element)

查

get(index);

subList(begin,end)

ListIterator(); 迭代器

indexOf

subList

用Iterator和ListIterator(是Iterator的子接口)进行数据的操作

其中，如果对Iterator进行操作会发生报错，因为在迭代时，不能通过创建集合对象来操作集合

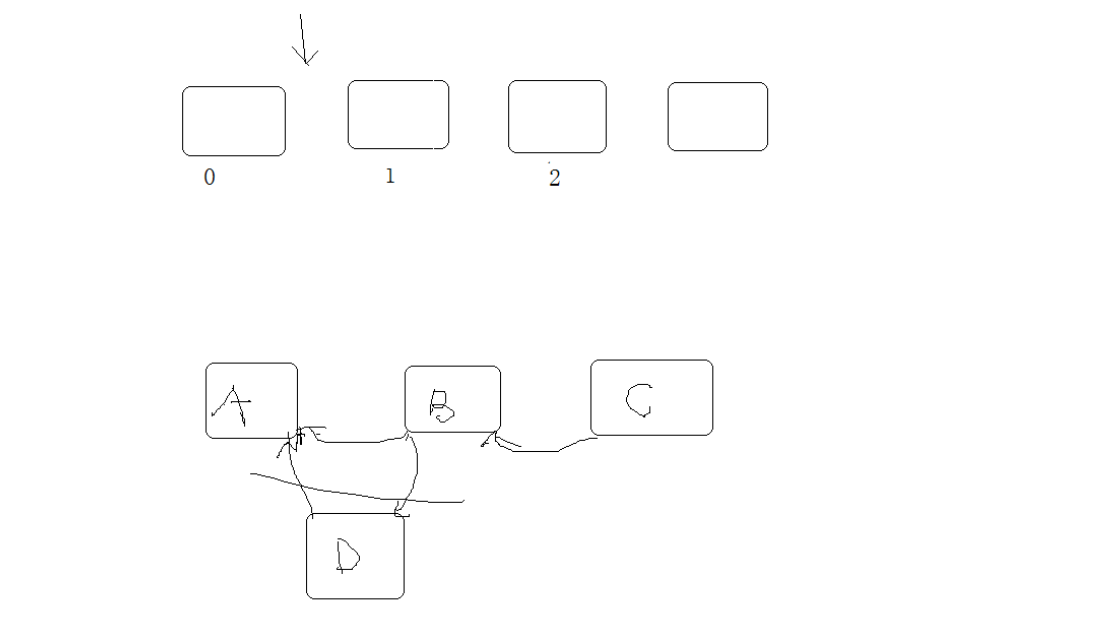
发生的报错类型为：Exception in thread "main" java.util.ConcurrentModificationException

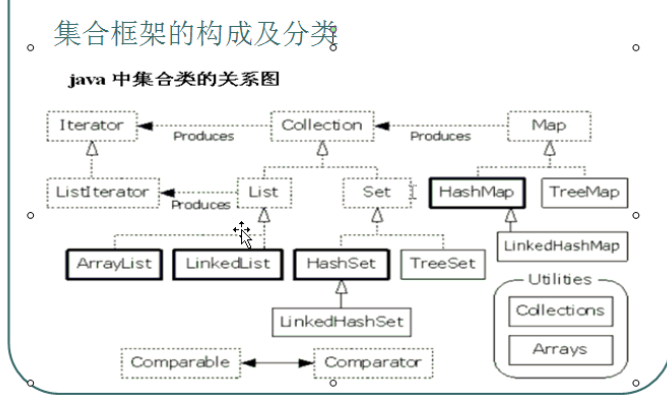
报错的原因：迭代器在用到集合时，同时又用到集合的功能来操作元素的话可能会存在安全隐患。

只用迭代器的方法操作，它的方法很少，所以引入一个ListIterator接口来扩展功能。

但是如果用ListIterator则可以通过创建集合对向来对集合进行操作。

链表数据结构与数组数据结构示意图：





1. **集合框架(LinkedList) 练习 对应175讲**

根据上节的内容，就是LinkedList所提供的各种方法来做一下练习。

用LinkedList这个类做一下 堆栈 队列 数据结构的东西

在这个练习中，构造函数的初始化可能还存在着一些遗忘的问题。

对应的博客复习：https://blog.csdn.net/aerchi/article/details/90759744

堆栈：先进后出。

队列：先进先出 First in First out (FIFO)

这个例子中，用类来封装做？

为什么要封装呢？ 因为搞开发的时候自己定义方法，然后自己较熟悉这个东西到底是什么，是谁做的。

6.集合框架(ArrayList)练习1

需求：输出其中不重复的元素，并存在新的列表之中

用到的方法：contians() hasNext() next

整体的编程思路：通过迭代观察该元素是否已经包含在新的容器之中了，如果没有包含，则让其加入，如果有的话就继续循环直到循环完成 迭代器Iterator的hasNext() next()方法

这道题还有很大问题自己搞混了都。。。。 错误找到了，记录在错误17中。

7.集合框架(ArrayList)的练习 2 对应176讲

这个练习可以好好看一下！！！ 还没有搞出结果来。。 浪费很多时间了，先过了

将自定义对象作为元素存到ArrayList集合中，并取出重复元素。

ex：存人对象，同姓名同年龄，实为同一个人，为重复元素。

思路：1.对人描述，将数据封装进入对象。

2.定义容器，将人存入。

3.取出。

要点：定义一个Person类， 用到的方法： Iterator hasNext next

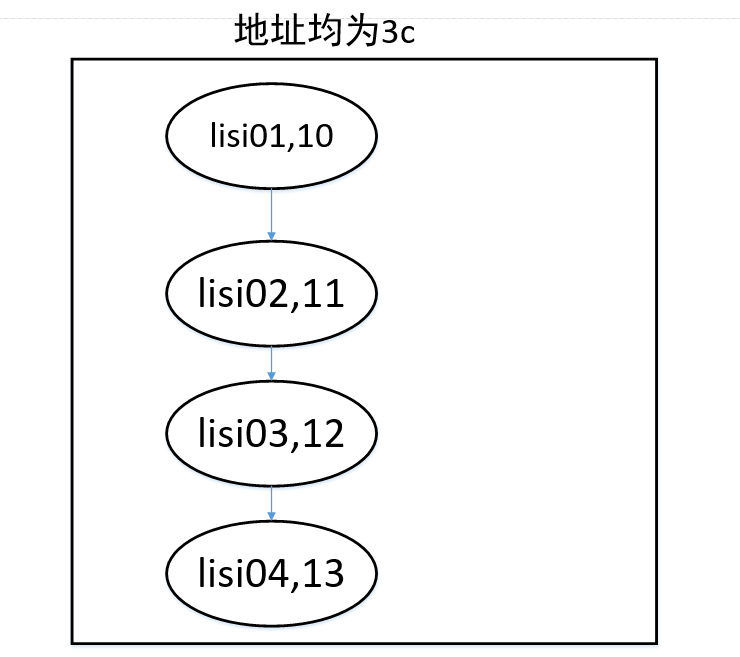
在去除重复元素之后还需要用迭代的方法取出元素

1. 哈希集合的特点： 对应177讲

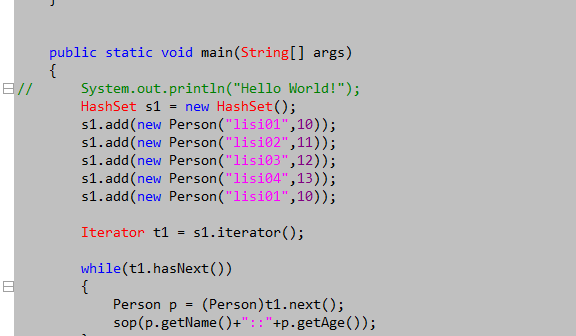
搞一下哈希集合：

特点：1.无序 2.元素不可重复。

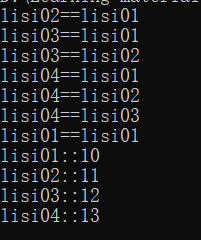
1. 集合框架(HashSet存储自定义对象) 对应178讲



对应的程序

：

对应的结果:



所以比较的过程为：lisi01进入，然后lisi02进入。然后进行比较，因为两个东西不相等，所以存储。Lisi03进入与lisi01比较，不相等，然后再与lisi02比较不相等，所以存储lisi03；lisi04也是同理。不同的是另一个lisi01和最先存储的lisi01已经相等，所以不需要存储，直接不让存储了。

|-- Set: 元素是无序的，(存入和取出的顺序不一定一致)，元素不可以重复。

|--HashSet:底层数据是哈希表。

HashSet 如何保证元素的唯一性？

是通过元素的两个方法：hashCode和equals来完成

如果元素的Hashcode相同，才会调用equals是否为true

如果元素的HashCode不同，不会调用equals。

HashTest 集合框架(HashSet)自定义存储对象

这是一个关于自定义存储对象的练习。

注意：对于hashSet来判断元素是否存在，以及删除元素依赖的是hashCode()以及equals()方法。这与ArrayList不同，ArrayList仅仅依赖的是equals方法。

equals和hashSet都是默认的方法，equals相等则不输出，不相等则输出。

注意，当定义的数据结构是Hashset时，equals这个方法需要复写。