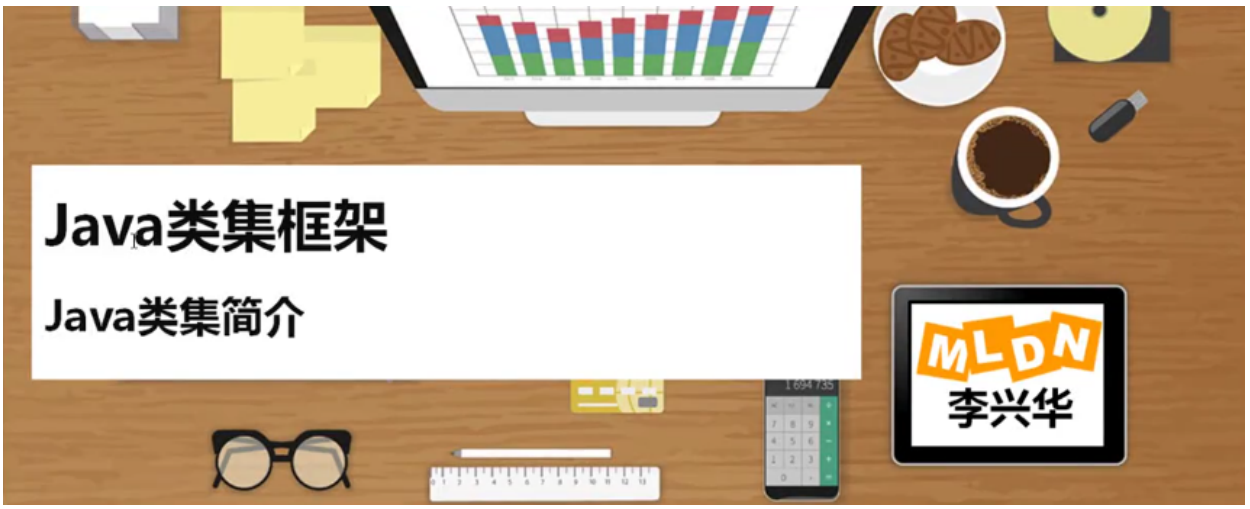

博客： https://www.cnblogs.com/HOsystem/p/14116443.html

2、具体内容

从JDK1.2开始Java引入了类集开发框架，所谓的类集指的就是一套动态对象数组的实现方案，在实际的开发之中没有任何一项的开发可以离开数组，但是传统的数组实现起来非常的繁琐，而且长度是其致命伤，正是因为长度的问题，所以传统的数组是不可能大范围使用的，但是我们的开发又不可能离开数组，所以最初就只能依靠一些数据结构来实现动态的数组处理，而其中最为重要的两个结构：链表、树，但是面对这些数据结构的实现又不得不面对如下的一些问题：

·数据结构的代码实现困难，对于一般的开发者是无法使用的；

·对于链表或二叉树当进行更新处理的时候的维护是非常麻烦的；

·对于链表或二叉树还需要尽可能保证其操作的性能。

正是因为这样的原因，所以从JDK1.2开始Java引入了类集，主要就是对常见的数据结构进行完整的实现包装，并且提供有了一系列的接口与实现子类来帮助用户减少数据结构所带来的开发困难，但是最初的类集实现由于Java本身的技术所限，所以对于数据的控制并不严格，全部采用Object类型进行数据接收，而在JDK1.5之后由于泛型技术的推广，所以类集本身也得到了了良好的改进，可以直接利用泛型来保存相同类型的数据，并且随着数据量的不断增加，从JDK1.8开始类集中的实现算法也得到了良好的性能提升。

在整个类集框架里面提供有如下的几个核心结构：Collection、List、Set、Map、Iterator、Enum
eration、Queue、ListIterator。

2、具体内容

java.util.Collection是单值集合操作的最大的父接口，在改接口之中定义有所有的单值数据的处理操作，这个接口之中定义有如下的核心操作方法：

No	方法名称	类型	描述
01	public boolean add(E e)	普通	向集合保存数据
02	boolean addAll(Collection<? extends E> c)	普通	追加一组数据
03	public void clear()	普通	清空集合,让根节点为空,同时执行GC处理
04	public boolean contains(Object o)	普通	查询数据是否存在,需要equals()方法支持
05	public boolean remove(Object o)	普通	数据删除，需要equals()方法支持
06	public int size()	普通	获取数据长度
07	public Object[] toArray()	普通	将集合变为对象数组返回
08	public Iterator<E> iterator()	普通	将集合变为Iterator接口

在进行集合操作的时候有两个方法最为常用【数据添加】add()、【数据输出】iterator()，在JDK1.5版本以前，Collection只是一个独立的接口，但是从JDK1.5之后提供有了Iterable父接口，并且从JDK1.8的之后针对于Iterable接口也得到了一些扩充。另外，在JDK1.2~JDK1.4的时候里面如果要进行稽核的使用往往会直接操作Collection接口，但是从JDK1.5时代开始更多的情况下选择的都是Collection的两个子接口：允许重复的List子接口、不允许重复的Set子接口；

Collection接口