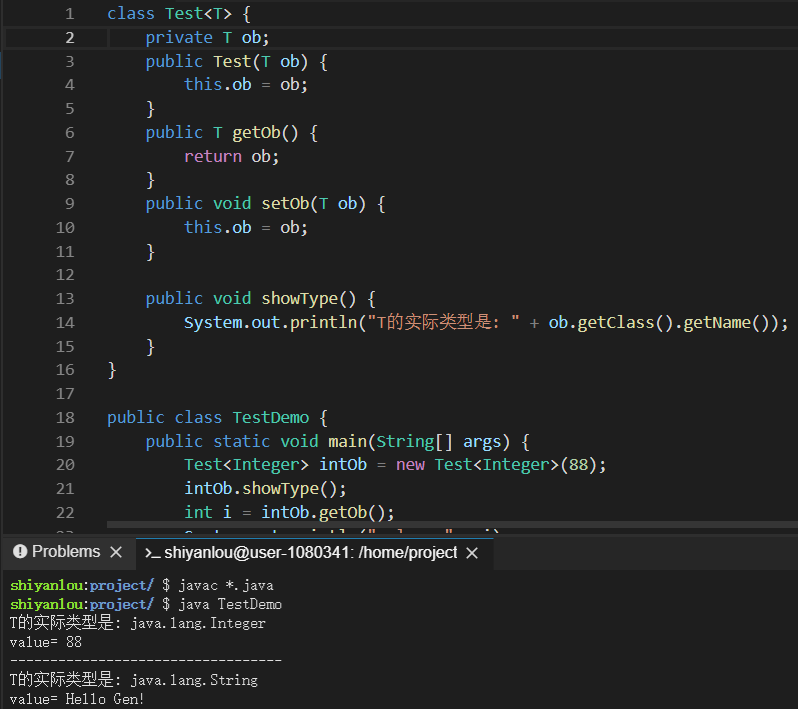
20190526学习心得

1. 泛型

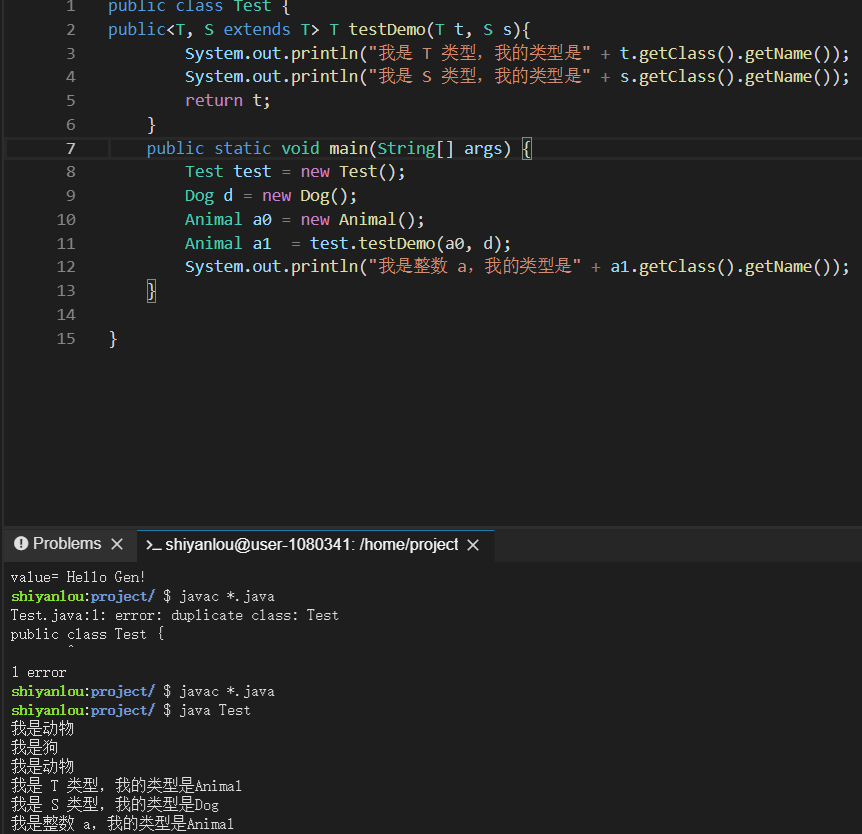
泛型即参数化类型，也就是说数据类型变成了一个可变的参数，在不使用泛型的情况下，参数的数据类型都是写死了的，使用泛型之后，可以根据程序的需要进行改变

定义泛型的规则

1. 只能是类类型，不能是简单数据类型
2. 泛型参数可以有多个
3. 可以用使用 extends 语句或者 super 语句 如<T extends superClass>表示类型的上界，T 只能是 superClass 或其子类， <K super childClass>表示类型的下界，K 只能是 childClass 或其父类
4. 可以是通配符类型，比如常见的 Class<?>



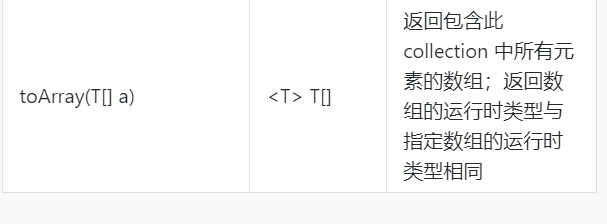
1. 使用泛型方法



1. Collection





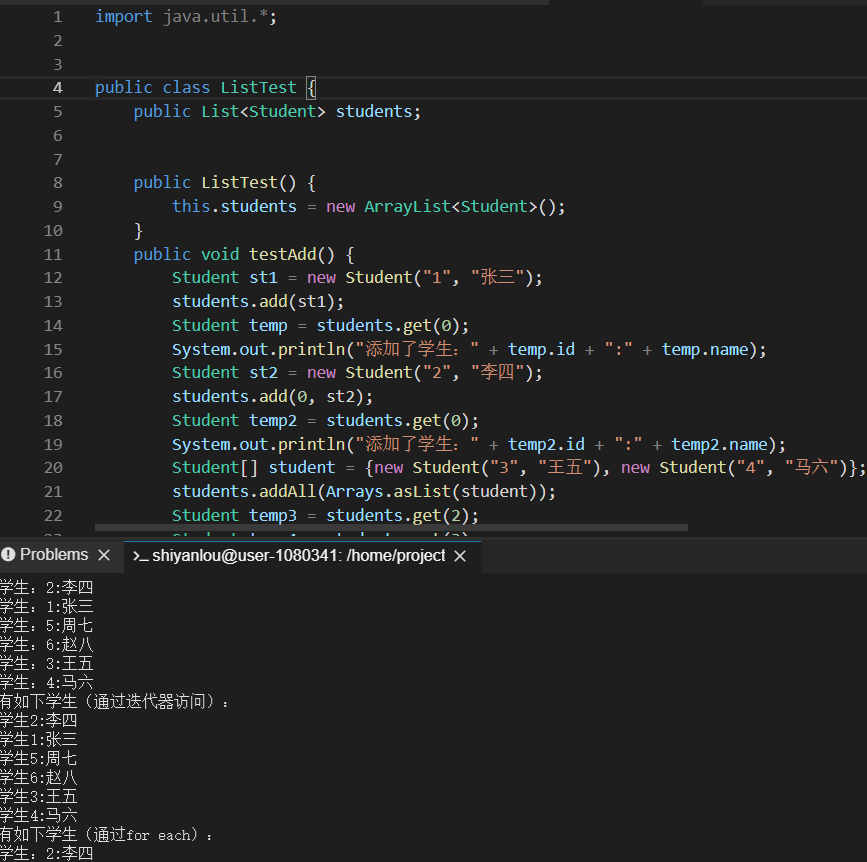


1. List



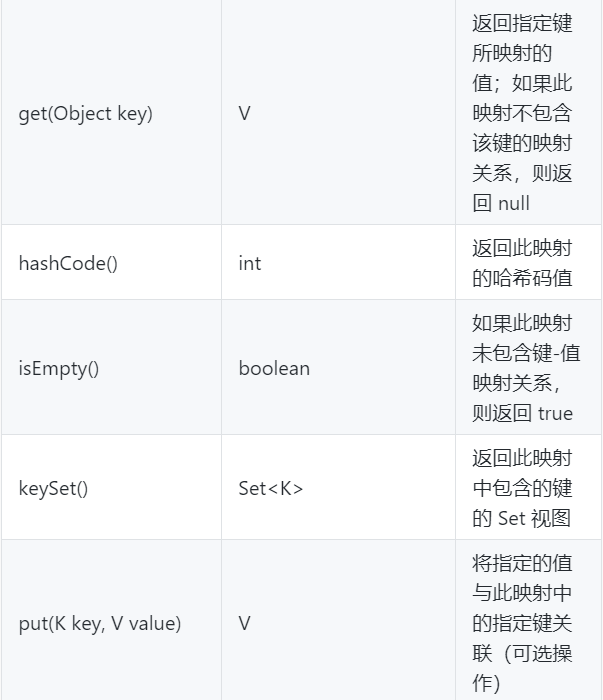


1. ArrayList



1. Map

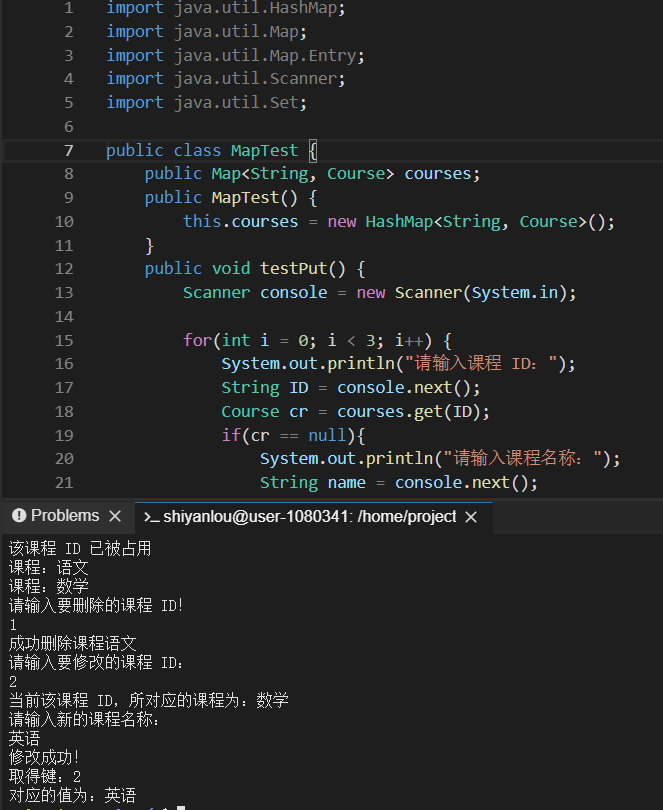






1. HashMap

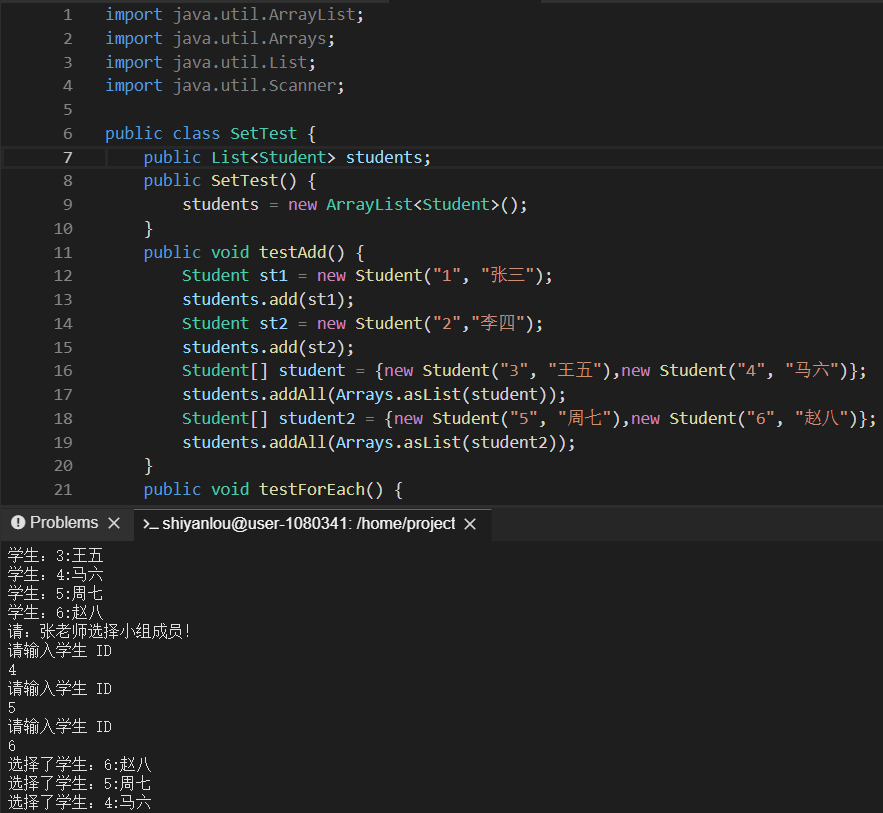
HashMap 是基于哈希表的 Map 接口的一个重要实现类



1. Set与HashSet

Set 接口也是 Collection 接口的子接口，它有一个很重要也是很常用的实现类——HashSet，Set 是元素无序并且不包含重复元素的 collection（List 可以重复），被称为集

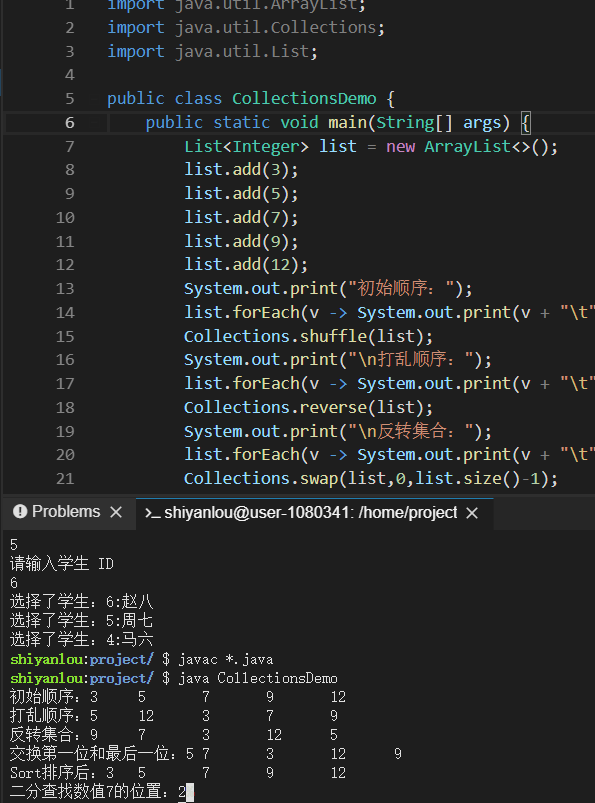
HashSet 由哈希表（实际上是一个 HashMap 实例）支持。它不保证 set 的迭代顺序；特别是它不保证该顺序恒久不变



1. Collections

java.util.Collections 是一个工具类，他包含了大量对集合进行操作的静态方法





1. 算法

排序算法

1. 插入排序

排序过程如下

1. 从第一个元素开始，该元素可以认为已经被排序

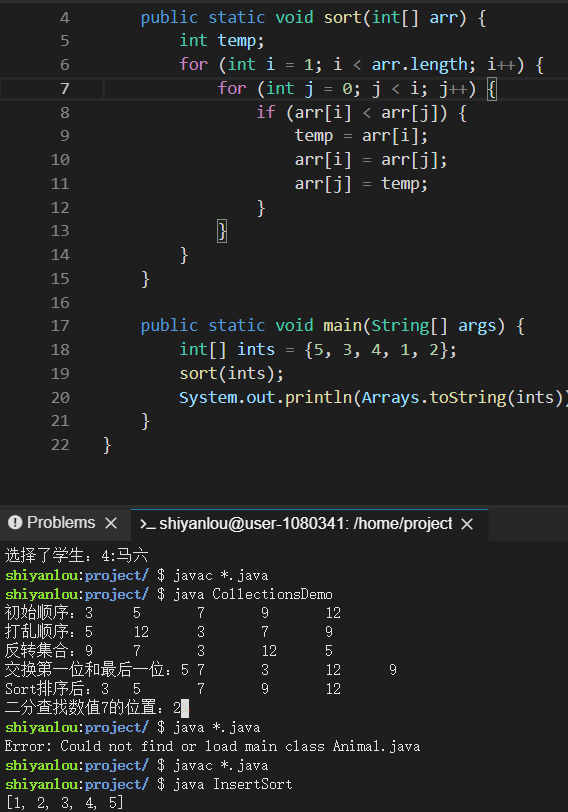
2. 取出下一个元素，在已经排序的元素序列中从后向前扫描

3. 如果该元素（已排序）大于新元素，将该元素移到下一位置

4. 重复步骤 3，直到找到已排序的元素小于或者等于新元素的位置

5. 将新元素插入到该位置后

6. 重复步骤 2~5



1. 冒泡排序

1. 比较相邻的元素。如果第一个比第二个大，就交换他们两个

2. 对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。这步做完后，最后的元素会是最大的数

3. 针对所有的元素重复以上的步骤，除了最后一个

4. 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较



1. 归并排序

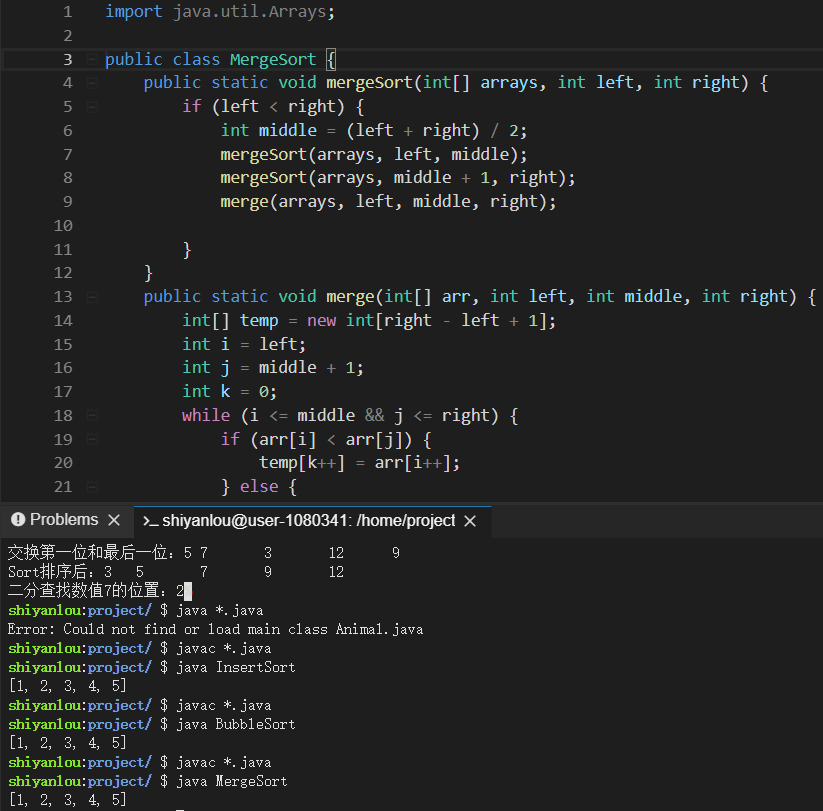
1. 申请空间，使其大小为两个已经排序序列之和，该空间用来存放合并后的序列

2. 设定两个指针，最初位置分别为两个已经排序序列的起始位置

3. 比较两个指针所指向的元素，选择相对小的元素放入到合并空间，并移动指针到下一位置

4. 重复步骤 3 直到某一指针到达序列尾

5. 将另一序列剩下的所有元素直接复制到合并序列尾



1. 快速排序

1. 从数列中挑出一个元素，称为“基准”

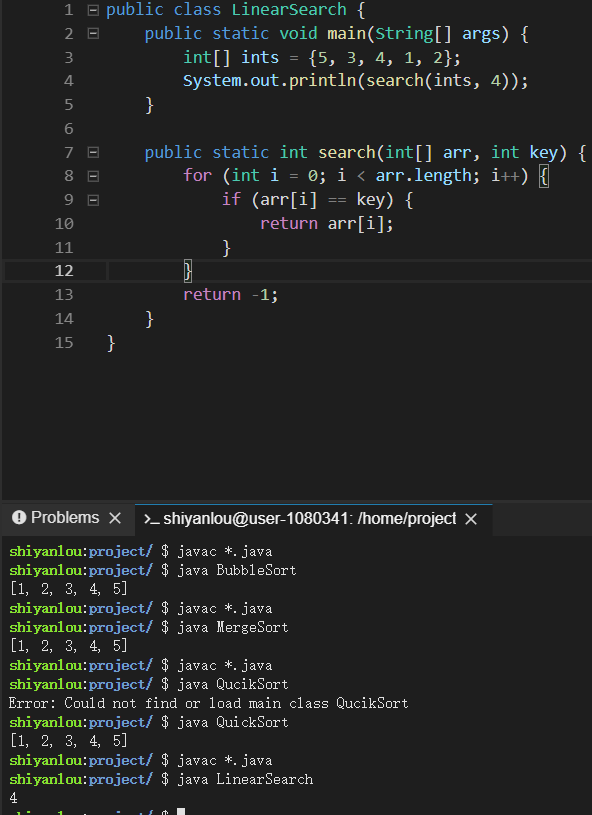
2. 重新排序数列，所有比基准值小的元素摆放在基准前面，所有比基准值大的元素摆在基准后面（相同的数可以到任何一边）。在这个分割结束之后，该基准就处于数列的中间位置。这个称为分割操作

3. 递归地把小于基准值元素的子数列和大于基准值元素的子数列排序

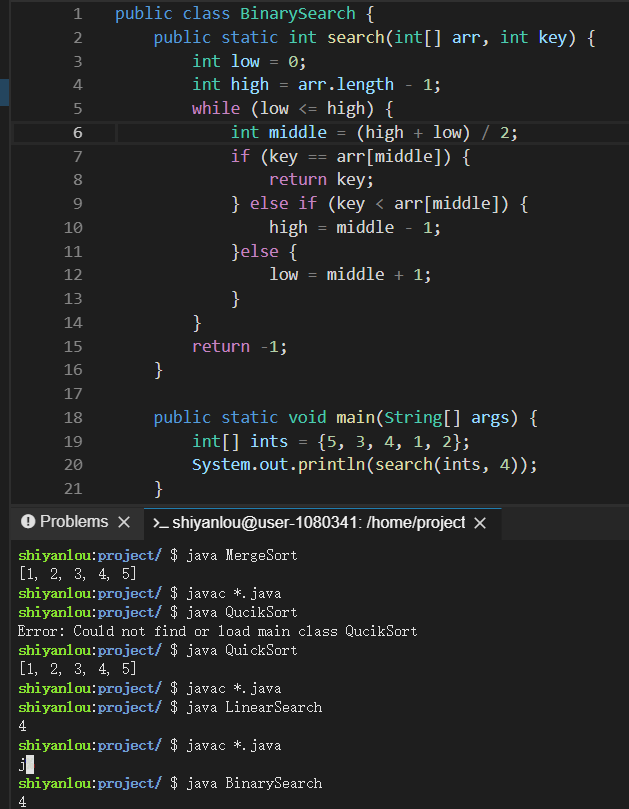


搜索算法

1. 线性搜索



1. 二分查找



明日计划java异常