**Java学习Day16**

1. **集合（Map概述） 对应day1,2**

Map集合： 该集合存储键值对，一对一对往里存，而且要保证键的唯一性;类似于python的字典

1.添加

put(K key, V value)

putAll(Map<? extends K,? extends V>m)

2.删除

clear()

remove(Object key)

3.判断

containsValue(Object value)

containsKey(Object key)

isEmpty()

4.获取

get(Object key)

size()

values()

这里补充一下Map集合的三种结构：Hash Tree Hashtable

2.集合(Map子类集合对象特点) 对应day3,4,5

|--Hashtable:底层是哈希表数据结构，不可以存入null值和null键。该集合是线程同步的，jdk1.0效率低

|--HashMap：底层是哈希表数据结构。允许使用null值null键，该集合是不同步的。jdk1.2效率高。

|--TreeMap：底层是二叉树数据结构。线程不同步。可以用于给map集合中的键进行排序。

和Set很像，其实Set的底层就是用了Map集合。

2.集合Map-KeySet

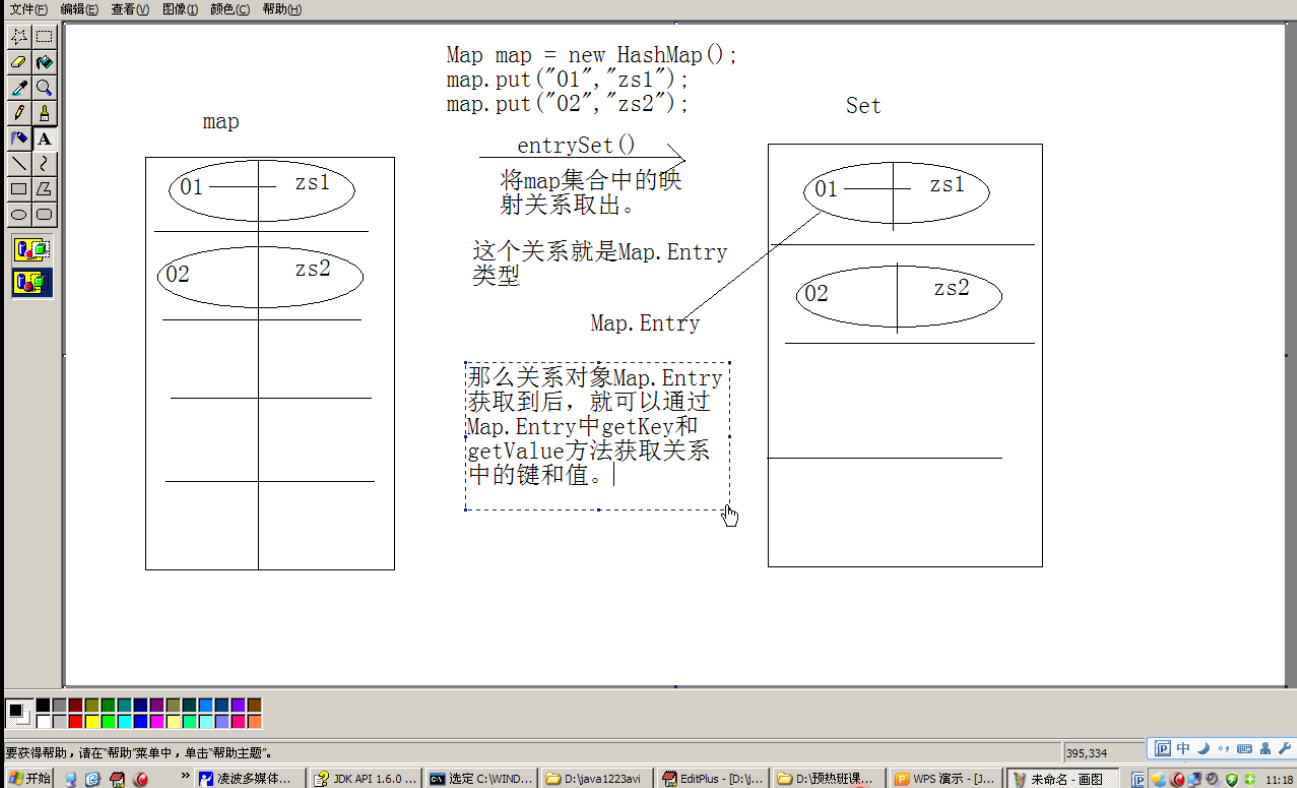
map集合的两种取出方式

1. Set <k> keySet() 将map中的所有健存入到Set集合。因为Set具有迭代器，然后根据迭代器

取出相应的键，然后根据键可以通过Map的get方法得到对应的值

2. Set<Map.Entry<k,v>> entrySEt: 将map集合中的映射关系存入到了Set集合中，而这个关系的数据类型就是Entry

返回一个 Set的关键视图包含在这个Map。



3.集合(Map练习) 对应day06

练习：每一个学生都有对应的归属地，

学生Student,地址：String

学生属性：姓名，年龄。

注意：姓名和年龄相同时为同一个学生，保证学生的唯一性。

保证学生唯一性的方法：可以想到在学生类中重写hashCode()和equals()方法。

1. 集合（Map练习2）对应day07 08

需求：对学生对象的年龄进行升序。

因为数据是以键值对形式存在的。

所以要使用排序的Map集合，TreeMap。

注意：如果排序的方法改变的话，也就是升序和降序这两种方式，那么只要改变比较器中对应的s1和s2的位置即可。

1. 集合 TreeMap练习，统计字母所出现的次数 对应day08

int和Integer的区别

1、Integer是int的包装类，int则是java的一种基本数据类型

2、Integer变量必须实例化后才能被使用，而int变量不需要

3、Integer实际是对象的引用，当new一个Integer时，实际上是生成一个指针指向此对象；而int则是直接存储数据值 。

4、Integer的默认值是null，int的默认值是0。

练习： 获取该字符串中各个字母出现的次数；

希望打印的结果为： {a:1,b:2.......} 以此类推

通过结果可以发现每个字母都有一个对应的次数，

说明字母与次数之间存在着一一对应的映射关系。

因此我们可以使用map集合。

为什么使用map集合？

当数据之间存在着映射关系时，就可以使用map集合。

思路：

1.将字符串转换为字符数组，因为要对每一个字母进行操作。 toCharArray() 将此字符串转换为一个新的字符数组。

2.定义一个map集合，因为打印结果的字母有序，所以使用treemap集合

3.遍历数组。 相当于py中的那个思路

用get方法括号中加键得到对应的值，并且如果值为空的话可以进行操作。

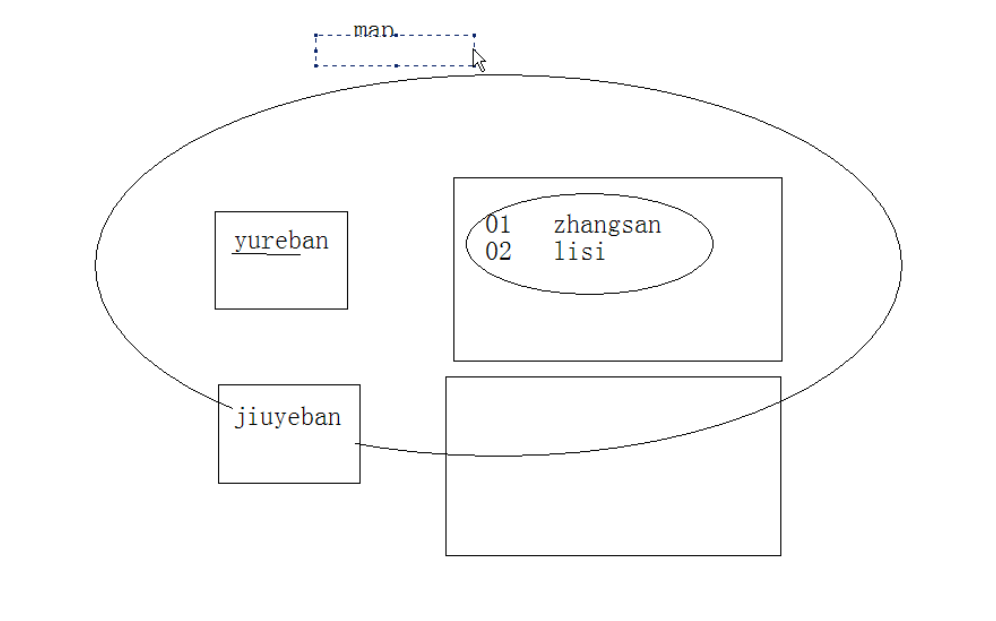
4.将map数据变成指定的字符串打印，这里就先打印一个map集合即可。

键是对应的字母，值时出现的次数。

1. Map扩展 对应day09

这是一个比较综合的练习

这个是当值是一个集合时的示意图：



Map扩展知识

map集合被使用是因为具备扩展关系

"yureban" "01" "zhangsan"

"yureban" "02" "lisi"

条件改变为学生对象 ：则里面没有对应的映射关系

"yureban" Student("01" "zhangsan");

"yureban" Student("02" "lisi");

这时用的应该是List列表类型或者Set类型。 大范围是一个映射，小范围是一个list

"jiuyeban" "01" "wangwu"

"jiuyeban" "02" "zhaoliu"

这个是当值是 Student对象时的集合图

