3.1 顺序容器

容器是现代程序设计非常基础而重要的手段。

所谓容器，就是“放东西的东西”。数组可以看作是一种容器，但是数组的元素个数一旦确定就无法改变，这在实际使用中是很大的不足。一般意义上的容器，是指具有自动增长容量能力的存放数据的一种数据结构。在面向对象语言中，这种数据结构本身表达为一个对象。所以才有“放东西的东西”的说法。

Java具有丰富的容器，Java的容器具有丰富的功能和良好的性能。熟悉并能充分有效地利用好容器，是现代程序设计的基本能力。

我们首先学习的是顺序容器，即放进容器中的对象是按照指定的顺序（放的顺序）排列起来的，而且允许具有相同值的多个对象存在。

ArrayList 类

ArrrayList<String> notes= new ArrayList<String>();

notes.add()； 只传入值，不传入对象本身，修改对象不会影响notes内部的值

notes.size()； 返回notes的长度

notes.get(index)； 返回notes第index位的值

notes.remove(index)； 移除notes第index位的值

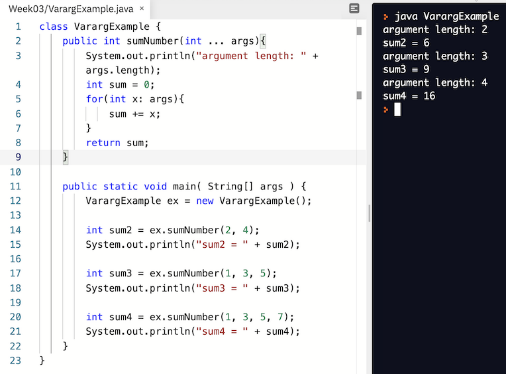
notes.toArray(a)； 将notes对象中的数据导入 a 数组中

3.2 对象数组

对象构成的数组，进行for each循环，循环变量为对象的管理者，即对循环变量进行操作，会影响数组内的对象。

Java本身支持任意数量参数的函数。

方法：在数据类型后添加“...”。



3.3 集合容器

集合就是数学中的集合的概念：所有的元素都具有唯一的值，元素在其中没有顺序。

HashSet<String> s = new HashSet<String>();

只要在类中定义成员函数 public String toString() { return …}，

则System.out.println(用该类创造的对象)，会直接返回成员函数toString()返回的值。

3.4 散列表 (Hash)

传统意义上的Hash表，是能以int做值，将数据存放起来的数据结构。Java的Hash表可以以任何实现了hash()函数的类的对象做值来存放对象。

Hash表是非常有用的数据结构，熟悉它，充分使用它，往往能起到事半功倍的效果。

public class coin{

private HashMap<Integer, String> = coinnames new HashMap<integer, String>;

（此处不能使用int, double等基础类型，需要放入对象，而Integer为int的包裹类型，可以作为类的对象放入。）

public Coin(){

coinnames.put(1, “penny”);

coinnames.put(10, “dime”);

coinnames.put(25, “quarter”);

coinnames.put(50, “half-dollar”);

}

public String getName(int amout){

if ( coinname.containKeys(amount) )

return coinname.get(amount);

else

return “NOT FOUND”;

}

可以直接println(coinnames); (HashMap有toString 功能)

遍历键值对：

for (Integer k: coinnames.keySet() ) {

String s = coinnames.get(k);

}