

ArrayList

ArrayList可通过Add(Object value)添加数据，
且因为参数是Object，所以不管是什么类型都可以往里放，随便放。
但取数据时就需要先判断类型，然后进行强制转换，因此不太方便。

一、增删改查

1.增

分别使用Add和AddRange添加单个和集合元素

```
ArrayList list = new ArrayList();

//向集合中添加单个元素
list.Add(3.14);
list.Add(true);
list.Add("张三");
list.Add(5000m);    //(m表示decimal类型)
//若使用Add添加集合元素，则在输出时需要转换成对应类型(因为此时集合中元素是Object型)
//如int[]型转成int[]型才能输出，class类转成class类输出，否则直接输出的是该元素的命名空间
//如System.Int32[]、ArrayListPractice.Human等

//向集合中添加集合或数组元素(不包括对象)
list.AddRange(new int[] {1,2,3,4,5,5,5,5});
list.AddRange(list);

for (int i = 0; i < list.Count; i++)
{
    System.Console.WriteLine(list[i]);
}
```

2.删

```
list.Clear();           //清空所有元素
list.Remove("张三");    //删除单个元素，写谁名字就删谁
list.RemoveAt(1);       //删除指定下标的元素
list.RemoveRange(0,3);  //删除指定范围内的元素(从0开始，一共删三个)
```

3.改

```
list.Sort();            //将所有元素升序排列(必须是同类且可比较的数据,如int和double都不能比较)
```

```
list.Reverse();           //将所有元素颠倒排列
list.Insert(1,"这是一个插入的元素");    //在指定位置插入单个元素
list.InsertRange(3,new string[]{"李四","王麻子"});    //在指定位置插入一个集合或数组
```

4.查

```
//遍历输出集合所有元素
for (int i = 0; i < list.Count; i++)
{
    System.Console.WriteLine(list[i]);
}

//判断是否包含某个指定的元素
bool b = list.Contains(1);
//举例:
if(!list.Contains("李四")){
    list.Add("李四");    //没有则添加
}else{
    System.Console.WriteLine("已包含李四");
}
```

二、ArrayList的长度问题

使用.Count查询集合中**实际**包含的元素个数,设为a

使用.Capacity查询集合中**可以**包含的元素个数,设为b

当a = 0时, b = 0; 当a ∈ [1,4] 时, b = 4

当a > b时,会自动扩容, b = b*2,直到存放完所有元素

```
ArrayList list = new ArrayList();
list.Add("s");
System.Console.WriteLine(list.Count);    //结果为1
System.Console.WriteLine(list.Capacity);    //结果为4

//每当.Count超过2*n(n > 1 )次幂时, .Capacity就会扩容到2*(n+1)次幂, 以确保能存放完所有元素
ArrayList list2 = new ArrayList();
for (int i = 0; i < 18; i++)
{
    list2.Add("s");
}
System.Console.WriteLine(list2.Count);    //结果为18
System.Console.WriteLine(list2.Capacity);    //结果为32
```

三、练习

写一个长度为10的集合, 要求在里面随机地存放10个数字(0-9), 但是要求所有数字不重复

```
ArrayList list = new ArrayList();
Random r = new Random();
for (int i = 0; i < 10; i++)
{
    int rNumber = r.Next(0,10);
    if(!list.Contains(rNumber)){    //若集合中没有该随机数则添加
        list.Add(rNumber);
    }else{    //若集合中有该随机数则不添加，本次循环作废，倒退再循环一次
        i--;
    }
}
for (int i = 0; i < list.Count; i++)
{
    System.Console.WriteLine(list[i]);
}
```