**ArrayList 集合 非静态类**

**因为ArrayList是非静态类，所以我们使用的时候需要创建一个对象实例化：**



**注意：**

1. **我们在集合里面装的元素全都是装的一个对象，比如说装1，那么就是int类型的对象，装”a”就是string类型的对象，因为集合list是object类，而int和string都属于object类的子类，根据里式转换第一条，子类对象可以装在父类里面，也就是说装进去的都是list的子类，所以当我们装进去了一个int 对象，但是取出来的是object类型，根据里式转换第二条，我们可以进行强转，转换成存进去的类型，再进行使用**

**假设装的是：**

**list.Add(1); ==>object list = new int();**

**取出来的时候自然是object类型**

**所以需要根据里式转换第二条强转才能使用：int b = (int)list[i] +1；**

**2. 当我们为集合添加元素后，我们可以通过list[下标]来访问该下标中元素的值，也可以通过list[下标] = value来改变该下标中元素的值**

1. **list中的方法：**
2. List.Add() 添加单个元素**（对象)**

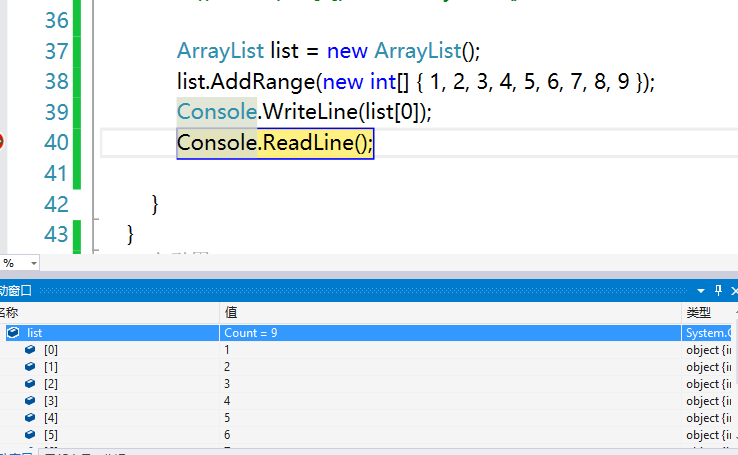
如图：



**（因为list是一个object类，为一起切的父类，里面装的都是对象，所以我们通过访问下标的输出的方式输出的是一个 对象，我们想使用的时候就要进行强制转换！里式转换原则）**

1. list.AddRange(); 添加集合元素数组和集合**(对象)**

**当放数组的时候会把该数组中的每个元素都变成 集合 中的 下标，所以当我们像访问下图的数组中的元素的时候需要输出list[0]~list[9]，而不是只输出一个list[0]**



1. list.Clear(); 清空所有元素(对象)
2. list.Remove(object value); 删除单个元素**(对象)**，写谁删谁,，删除元素后面的元素的下标往前增
3. list.RemoveAt(int index); 删除指定下标的单个元素**(对象)**，下标从0开始，删除元素后面的元素的下标往前增
4. list.RemoveRange(int index,int count); 删除由指定下标开始长度为count中的元素**(对象)**，index下标从0开始，长度从count从1开始，删除元素后面的元素的下标往前增

7. list.Sort(); 升序排列 集合中的元素**(对象)**类型要相同

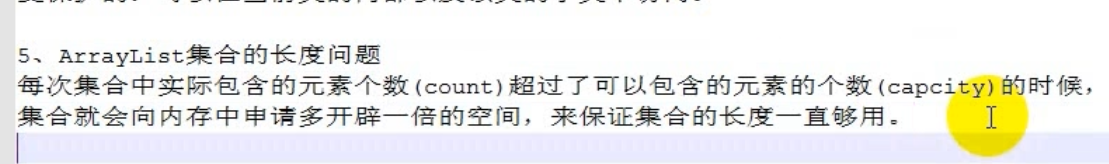
8. list.Reverse(); 集合反转，也就是把下标为0的编程最后一个下标，以此类推进行反转

9. list.Insert(int index,插入的元素)；在指定下标插入个元素**(对象)**，原先在此下标的元素**(对象)**，下标往后排

10. list.InsertRange(int index ,插入的集合)；在指定位置的下标插入一个集合**(对象)**，原先在此位置的元素**(对象)**的下标往后排

11. bool bol = list.Contains(object value); 判断该集合中是否存在该元素，返回bool

1. **集合的长度问题**



list.Count：获取这个集合中**目前所包含的元素的个数**

list.Capcity：获取这个集合中**可以包含的元素的个数**