华北电力大学

**实 验 报 告**

|

|

**实验名称** 队列与多线程

**课程名称**  数据结构综合实践

|

|

专业班级： 信息2202 学生姓名徐梓乔

学 号： 220221100327 成 绩：

|  |
| --- |
| 一、实验内容  排队叫号软件的具体操作流程为：顾客取服务序号，当顾客抵达服务大厅时，前往放置在入口处旁的取号机，并按一下其上的相应服务按钮，取号机会自动打印出一张服务单。单上显示服务号及该服务号前面正在等待服务的人数。服务员工呼叫顾客：服务员工只需按一下其柜台上呼叫器的相应按钮，则顾客的服务号就会按顺序的显示在显示屏上，并发出“叮咚”和相关语音信息，提示顾客前往该窗口办事。当一位顾客办事完毕后，柜台服务员工只需按呼叫器相应键，即可自动呼叫下一位顾客。  编写程序模拟上面的工作过程，主要要求如下：  程序运行后，当看到“请点击触摸屏获取号码：”的提示时，只要按回车键，即可显示“您的号码是：XXX，您前面有YYY位”的提示，其中XXX是所获得的服务号码，YYY是在XXX之前来到的正在等待服务的人数。  用多线程技术模拟服务窗口（可模拟多个），具有服务员呼叫顾客的行为，假设每个顾客服务的时间是5000ms，时间到后，显示“请XXX号到ZZZ号窗口！”的提示。其中ZZZ是即将为客户  服务的窗口号。  二、实验方法与步骤  public class CSeqBankQueue : CSeqQueue<int>, IBankQueue, IQueue<int>  {  public int Callnumber { get; private set; }  public CSeqBankQueue(int size)  : base(size)  {  Callnumber = 0;  }  public int GetCallnumber()  {  if (IsEmpty() && Callnumber == 0)  {  Callnumber = 1;  }  else  {  Callnumber++;  }  return Callnumber;  }  [SpecialName]  int IBankQueue.get\_MaxSize()  {  return base.MaxSize;  }  }  链队列:  public class LinkBankQueue : LinkQueue<int>, IBankQueue, IQueue<int>  {  public int Callnumber { get; private set; }  public int MaxSize { get; }  public int GetCallnumber()  {  if (IsEmpty() && Callnumber == 0)  {  Callnumber = 1;  }  else  {  Callnumber++;  }  return Callnumber;  }  public LinkBankQueue()  {  MaxSize = 0;  Callnumber = 0;  }  }  程序上,建立输入的容纳人数所相应的队列:  IBankQueue bankQueue = null;  Console.WriteLine("请选择存储结构的类型：1.顺序队列 2.链队列");  string text = Console.ReadLine();  string text2 = text;  if (!(text2 == "1"))  {  if (text2 == "2")  {  bankQueue = new LinkBankQueue();  }  }  else  {  Console.Write("请输入队列可容纳人数：");  int size = Convert.ToInt32(Console.ReadLine());  bankQueue = new CSeqBankQueue(size);  }  int num = 3;  ServiceWindow[] array = new ServiceWindow[num];  Thread[] array2 = new Thread[num];  for (int i = 0; i < num; i++)  {  array[i] = new ServiceWindow();  array[i].BankQ = bankQueue;  array2[i] = new Thread(array[i].Service);  array2[i].Name = (i + 1).ToString();  array2[i].Start();  }  实现排队取号:  if (bankQueue != null && (bankQueue.Length < bankQueue.MaxSize || text == "2"))  {  int callnumber = bankQueue.GetCallnumber();  Console.WriteLine("您的号码是：{0}，你前面有{1}位，请等待！", callnumber, bankQueue.Length);  bankQueue.EnQueue(callnumber);  }  else  {  Console.WriteLine("现在业务繁忙，请稍后再来！");  }  Console.WriteLine();  三、实验结果    四、结论与总结  结论:运用相关队列知识可以设计出模拟银行叫号的程序,通过顺序队列和链队列两种不同的队列组成叫号程序,采用多线程技术模拟服务窗口.  总结:队列是一个先进先出的线性表,能够实现数值插入队尾(入队),移除队首中的数据元素(出队),判断队中是否包含数据元素,获取队中实际包含数据元素的个数(得到队长),移除队中的所有数据元素(清空),获取队首元素. |
|  |