## 实验9 实现队列、栈的操作

### 实验目的

掌握集合类的使用

### 第一部分 上机训练

1、模拟食堂打饭的例子，理解队列模型

1）新建一个名为QueueDemo的控制台应用程序

2）新建一个名为Person的类，代码如下：

using System;

using System.Collections.Generic;

using System.Text;

namespace QueueDemo

{

public class Person

{

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

public Person(string name)

{

this.name = name;

}

public void BuyFood()

{

Console.WriteLine("轮到{0}为队的排头，{0}高兴地向阿姨指点着自己想要的饭菜。", name);

}

}

}

3）Program.cs的代码如下

using System;

using System.Collections.Generic;

using System.Text;

namespace QueueDemo

{

class Program

{

static void Main(string[] args)

{

Queue<Person> personQueue = new Queue<Person>();

//填充队列，此处使用了泛型，简化编程

Person person1 = new Person("张三");

Person person2 = new Person("李四");

Person person3 = new Person("王五");

personQueue.Enqueue(person1);

personQueue.Enqueue(person2);

personQueue.Enqueue(person3);

//排队轮流打饭

Person personToBuyFood = personQueue.Dequeue() as Person;

personToBuyFood.BuyFood();

StillInQueue(personQueue);

}

static void StillInQueue(IEnumerable<Person> myCollection)

{

IEnumerator<Person> myEnumerator = myCollection.GetEnumerator();

Console.Write("正在排队等待打饭的还有：");

while (myEnumerator.MoveNext())

Console.Write("{0}\t", myEnumerator.Current.Name);

Console.WriteLine();

}

}

}

### 第二部分 实战提高

采用弹夹上子弹时，最先上的子弹最后才能发射出去，结合MSDN，研究栈的特性，并使用栈模拟这一模型。



代码如下：

using System;

using System.Collections.Generic;

using System.Text;

namespace MySoftwareOnDotnetFramework.Comparable

{

class Program

{

static void Main(string[] args)

{

Stack<int> bullets = new Stack<int>();

Random random = new Random();//使用随机数模拟子弹编号

Console.WriteLine("子弹装填中...");

for(int i = 0; i < 10; i++)//装填10颗子弹

{

int temp = random.Next(0, 100);

Console.WriteLine("当前装填的子弹编号是：{0}" , temp);

bullets.Push(temp);

}

Console.WriteLine("\n装填完毕.\n开始发射.\n");

foreach(int i in bullets)

{

Console.WriteLine("编号为 {0} 的子弹射出去了！", i);

}

Console.ReadLine();

}

}

}

运行结果如下：

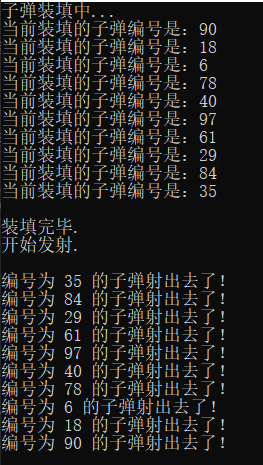


图9-1 创建子弹时的正序，弹栈的反序，说明了栈的后进先出特性