## 实验7 实现对集合类的操作

### 实验目的

掌握集合类的定义与初始化

掌握集合类的使用

掌握集合类的排序与反转

掌握foreach的用法

### 第一部分 上机训练

1、定义一个元素个数为5的整形数组，接收用户的输入，而后对该数组进行排序，输出排序后的结果。

1）建立一个名为ArrayDemo的控制台项目

2）双击Program.cs，输入如下代码：

using System;

using System.Collections.Generic;

using System.Text;

namespace ArrayDemo

{

class Program

{

static void Main(string[] args)

{

int[] number = new int[5];

for (int i = 0; i < 5; i++)

{

Console.Write("请输入第{0}个数：", i+1);

string input = Console.ReadLine();

number[i] = int.Parse(input);

}

Array.Sort(number);

Console.WriteLine("5个数排序后的结果为：");

for (int j = 0; j < 5; j++)

{

Console.Write("{0}\t", number[j]);

}

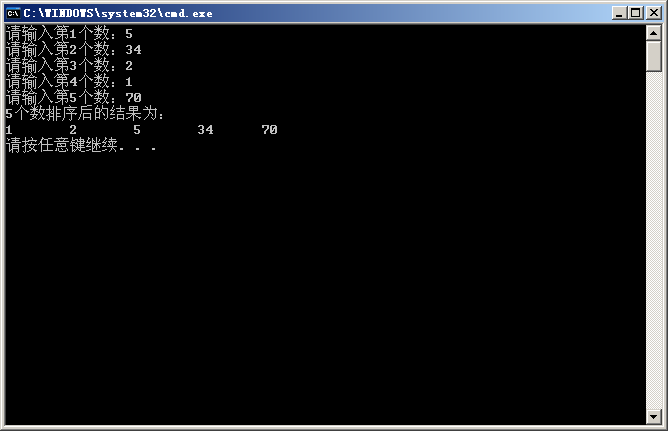
Console.WriteLine();

}

}

}

3）按Ctrl+F5运行，结果如下：



2、我院在计算学生期末总评成绩时，采用如下公式：

总评 =德育\*20% + 体育\*10%+学习\*70%

新建一个类，名称为Student,每个类中包含以下属性：姓名，德育分数，体育分数，学习成绩，总评成绩；并包含两个方法：计算总评成绩和显示总评成绩

在程序的Main方法中要求如下：

用户输入5名学生的德育分数，体育分数和学习成绩，计算其总评成绩，然后按总评成绩由大到小的顺序输出。

1）用VS.NET2005新建一个控制台项目，名称为Exam2

2）新建一个名为Student的类，代码如下：

using System;

using System.Collections.Generic;

using System.Text;

namespace Exam2

{

public class Student

{

private string \_sName;

private double \_mScore;

private double \_pScore;

private double \_sScore;

private double \_aScore;

public double MoralScore

{

get

{

return \_mScore;

}

set

{

\_mScore = value;

}

}

public double PhysicsScore

{

get

{

return \_pScore;

}

set

{

\_pScore = value;

}

}

public double StudyScore

{

get

{

return \_sScore;

}

set

{

\_sScore = value;

}

}

public string StudentName

{

get

{

return \_sName;

}

set

{

\_sName = value;

}

}

public double getAverageScore()

{

\_aScore = \_mScore \* 0.2 + \_pScore \* 0.1 + \_sScore \* 0.7;

return \_aScore;

}

public void displayInfo()

{

Console.WriteLine("{0}\t{1}\t{2}\t{3}\t{4}", \_sName, \_mScore, \_pScore, \_sScore, \_aScore);

}

}

}

3）Program.cs中代码如下：

using System;

using System.Collections.Generic;

using System.Text;

namespace Exam2

{

class Program

{

static void Main(string[] args)

{

Array arrStudent = Array.CreateInstance(typeof(Student), 5);

Array arrScore = Array.CreateInstance(typeof(double), 5);

for (int i = 0; i < 5; i++)

{

Student st = new Student();

Console.WriteLine("请输入学生的姓名:");

st.StudentName = Console.ReadLine();

Console.WriteLine("请输入学生{0}的德育成绩:", st.StudentName);

st.MoralScore = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("请输入学生{0}的体育成绩:", st.StudentName);

st.PhysicsScore = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("请输入学生{0}的学习成绩:", st.StudentName);

st.StudyScore = Convert.ToDouble(Console.ReadLine());

arrScore.SetValue(st.getAverageScore(), i);

arrStudent.SetValue(st, i);

}

Array.Sort(arrScore, arrStudent);

Array.Reverse(arrStudent);

Console.WriteLine();

Console.WriteLine("学生排名如下:");

Console.WriteLine("姓名\t德育\t体育\t学习\t总评");

for (int j = 0; j < 5; j++)

{

Student st = (Student)arrStudent.GetValue(j);

st.displayInfo();

}

}

}

}

### 实战提高

设计一个控制台应用程序，模拟管理车牌相关信息，

例如姓名和车牌号码，能够添加、修改、查找、删除、输出车牌信息。

（使用Dictionary<>类完成）

**代码和运行结果图请表示在下面：**

代码：

using System;

using System.Collections.Generic;

using System.Text;

namespace MySoftwareOnDotnetFramework.Shape

{

class Program

{

static void Main(string[] args)

{

Dictionary<string, string> dic = new Dictionary<string, string>();

while (true)

{

menu();

int op = int.Parse(Console.ReadLine());

switch (op)

{

case 1: add(dic);

break;

case 2: delete(dic);

break;

case 3: modify(dic);

break;

case 4: select(dic);

break;

case 5: display(dic);

break;

default: System.Environment.Exit(0);

break;

}

}

}

public static void add(Dictionary<string,string> dic)

{

Console.WriteLine("请输入车牌号");

string key = Console.ReadLine();

Console.WriteLine("请输入车主姓名");

string value = Console.ReadLine();

dic.Add(key, value);

Console.WriteLine("添加成功 :({0} , {1})", key, value);

Console.WriteLine();

}

public static void delete(Dictionary<string, string> dic)

{

Console.WriteLine("请输入车牌号");

string key = Console.ReadLine();

if (dic.ContainsKey(key))

{

dic.Remove(key);

Console.WriteLine("车牌信息 {0} 已成功删除", key);

Console.WriteLine();

}

else

{

Console.WriteLine("该车牌号不存在");

Console.WriteLine();

}

}

public static void modify(Dictionary<string, string> dic)

{

Console.WriteLine("请输入车牌号");

string key = Console.ReadLine();

if (dic.ContainsKey(key))

{

Console.WriteLine("请输入车主姓名");

string value = Console.ReadLine();

dic[key] = value;

Console.WriteLine("修改成功 :({0} , {1})", key, value);

Console.WriteLine();

}

else

{

Console.WriteLine("该车牌号不存在");

Console.WriteLine();

}

}

public static void select(Dictionary<string,string> dic)

{

Console.WriteLine("请输入车牌号");

string key = Console.ReadLine();

if (dic.ContainsKey(key))

{

string value = dic[key];

Console.WriteLine("查找成功 :({0} , {1})", key, value);

Console.WriteLine();

}

else

{

Console.WriteLine("该车牌号不存在");

Console.WriteLine();

}

}

public static void display(Dictionary<string,string> dic)

{

foreach(KeyValuePair<string,string> item in dic)

{

Console.WriteLine("车牌号 :{0} , 车主姓名 :{1}", item.Key, item.Value);

}

Console.WriteLine();

}

public static void menu()

{

Console.WriteLine("-----车牌管理系统-----");

Console.WriteLine("1 添加车牌号");

Console.WriteLine("2 删除车牌号");

Console.WriteLine("3 修改车牌号");

Console.WriteLine("4 查找车牌号");

Console.WriteLine("5 输出车牌信息");

Console.WriteLine("其他 退出");

Console.WriteLine("----------------------");

Console.WriteLine();

}

}

}

运行结果：

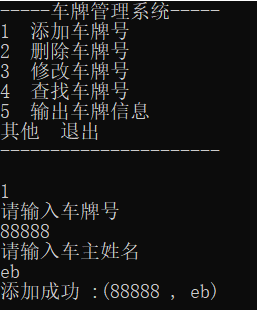


图7-1 添加车牌号

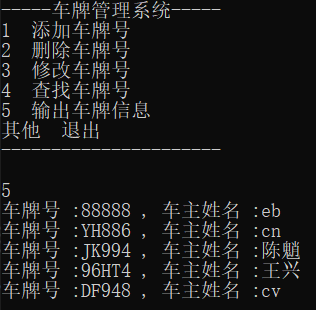


图7-2 展示所有车牌信息

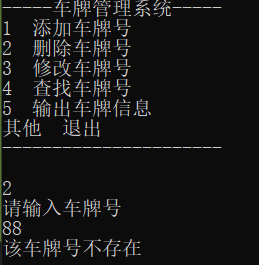


图7-3 删除车牌信息失败

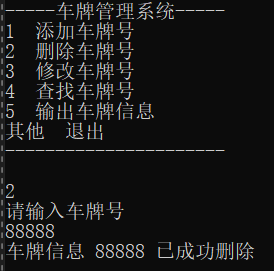


图7-4 删除车牌信息成功

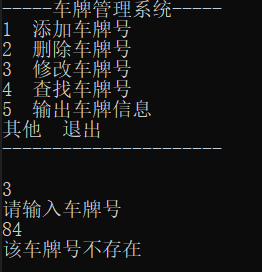


图7-5 修改车牌信息失败

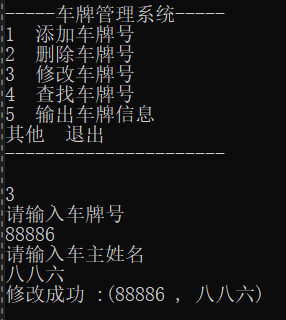


图7-6 修改车牌信息成功

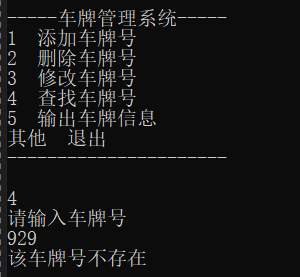


图7-7 查找车牌信息失败

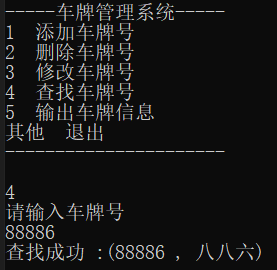


图7-8 查找车牌信息成功

### 第三部分 知识总结

|  |  |
| --- | --- |
| **学生总结** | 请将问题答案写在下面空白处 |
| 完成上机训练模块花费的时间 | 10分钟 |
| 完成实战提高模块花费的时间 | 45分钟 |
| 本次实验对你而言的难点 |  |
| 本次实验的收获 |  |
| 希望老师强化讲解的知识要点 |  |
| 希望老师帮助回答的问题 |  |
| 本次实验自评分数  （五分制） | 4.9 |