**学生管理系统**

2021110070 晁祥翔

1. 引入命名空间的介绍
2. using System ： System 命名空间是 .NET 中基本类型的根命名空间。 此命名空间包括表示由所有应用程序使用的基本数据类型的类，例如 Object（继承层次结构的根）、Byte、Char、Array、Int32 和 String。 在这些类型中，有许多与编程语言所使用的基元数据类型相对应。
3. using System. Collections.Generic : 集合（Collection）类是专门用于数据存储和检索的类。这些类提供了对栈（stack）、队列（queue）、列表（list）和哈希表（hash table）的支持。大多数集合类实现了相同的接口。集合（Collection）类服务于不同的目的，如为元素动态分配内存，基于索引访问列表项等等。这些类创建 Object 类的对象的集合。在 C# 中，Object 类是所有数据类型的基类。
4. using System.IO : 包含允许读取和写入文件和数据流的类型，以及提供基本文件和目录支持的类型。
5. 涉及C#知识点的介绍
6. 基础数据类型：

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 描述 | 范围 | 默认值 |
| Int | 32 位有符号整数类型 | -2,147,483,648 到 2,147,483,647 | 0 |
| Long | 64 位有符号整数类型 | -9,223,372,036,854,775,808 到 9,223,372,036,854,775,807 | 0L |
| bool | 布尔值 | True 或 False | false |

1. 引用类型 :

|  |  |  |
| --- | --- | --- |
| 类型 | 描述 | 技巧 |
| Object | 是 C# 通用类型系统（Common Type System - CTS）中所有数据类型的终极基类。Object 是 System.Object 类的别名。所以对象（Object）类型可以被分配任何其他类型（值类型、引用类型、预定义类型或用户自定义类型）的值。但是，在分配值之前，需要先进行类型转换。 | 当一个值类型转换为对象类型时，则被称为 **装箱**；另一方面，当一个对象类型转换为值类型时，则被称为 **拆箱**。 |
| string | 允许您给变量分配任何字符串值。字符串（String）类型是 System.String 类的别名。它是从对象（Object）类型派生的。 | 字符串（String）类型的值可以通过两种形式进行分配：引号和 @引号。 |
| Dictionary<> | Dictionary < TKey,TValue > 是一个泛型集合，它以不特定的顺序存储键值对。 | Dictionary<TKey, TValue> 存储键值对。  属于System.Collection.Generic命名空间。  实现 IDictionary <TKey，TValue>接口。  键必须是唯一的，不能为null。  值可以为null或重复。  可以通过在索引器中传递相关键来访问值，例如 myDictionary[key]  元素存储为 KeyValuePair <TKey，TValue> 对象。 |

1. 关键词
   1. **this**: 它可用于引用当前类实例变量;它可用于将对象作为参数传递的另一种方法；它可用于声明索引器。
   2. **var**: 可以赋予局部变量推断“类型”var 而不是显式类型。var 关键字指示编译器根据初始化语句右侧的表达式推断变量的类型。推断类型可以是内置类型、匿名类型、用户定义类型、.NET Framework 类库中定义的类型或任何表达式。
   3. **public**：修饰的类，可以在整个系统的任意地方调用，是完全公开的.
   4. **private**：关键字是一个成员访问修饰符。私有访问是允许的最低访问级别。私有成员只有在声明它们的类和结构体中才是可访问的。
   5. **using**： 关键字表明程序使用的是给定命名空间中的名称。
   6. **namespace** : 命名空间的设计目的是提供一种让一组名称与其他名称分隔开的方式。在一个命名空间中声明的类的名称与另一个命名空间中声明的相同的类的名称不冲突。
   7. **base**:访问基类的成员。用于从派生类中访问基类的成员，调用基类上已经被重写的方法。指定创建派生类实例时应调用的基类构造函数。
   8. **is** : 1.用于检查对象在运行时的类型是否与给定类型兼容。2.语法格式：expression is type3.当expression为非空并且可以被转化为type类型时，整个表达式返回true，否则返回false。4.is运算符不能被重载。
   9. New:关键字在C#中可以用作：1.运算符2.修饰符New运算符：用于在堆上创建对象和调用构造函数。New修饰符：用于隐藏基类成员的继承成员。
2. 设计过程
   1. 思维导图

图片包含 图表

描述已自动生成

* 1. 类 ： Class、Account、Admin、Course、Data、MyExemption、 People、Student、Teacher

图示

描述已自动生成

* 1. Class
     1. People: name, age
     2. Student : 继承People ; 属性有：学号，成绩，班级，账户；方法有：属性对应的get/set方法，重写ToString方法;
     3. Teacher : 继承People; 属性有：工号，班级，账户；方法有：属性对应的get/set方法，重写ToString方法；
     4. Class : 属性有：班级名，学生人数，学生（字典）；方法：属性对应的get/set方法，重写ToString方法，删除学生，求出班级平均值、极差、方差。
     5. Course：属性：成绩；方法有get/set方法
     6. Admin：属性：账户；方法有get/set方法
     7. Account：属性：id，password；方法有get/set方法
     8. MyException：继承于Exception，用于抛出自定的异常
     9. Data：属性：学生，老师，班级，instance；方法有 get/set方法，读取数据和保存数据
  2. 单例模式： private static Data Instance = null;

public static Data getInstance()

{

if (Instance == null)

{

Instance = new Data();

}

return Instance;

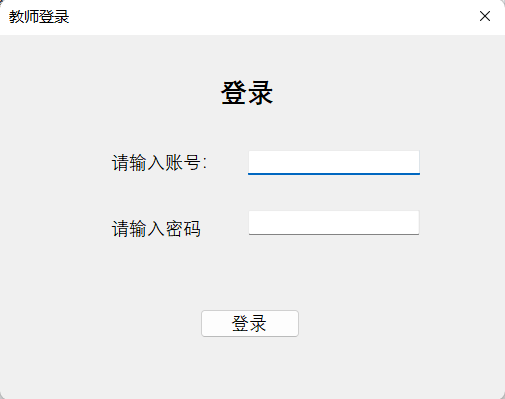
}

目的：保证数据只能创建一份。

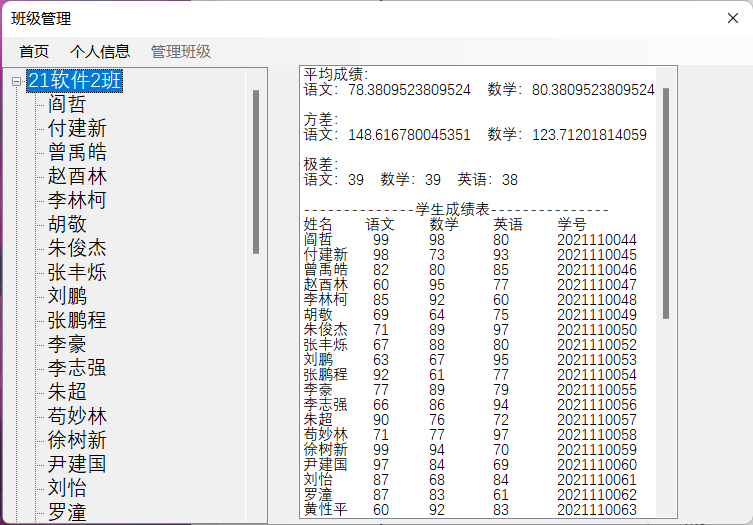
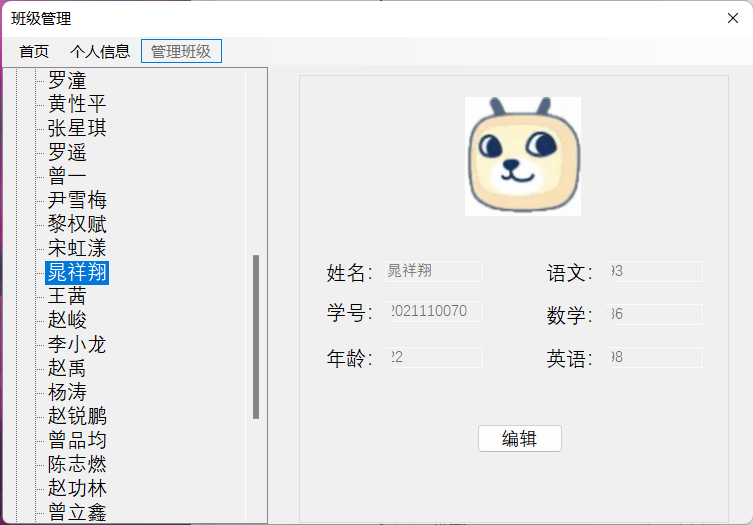
* 1. 静态代码块 ： 目的：创建Data类的时候将文本中的数据读取到内存中。

1. 界面设计过程  
   封面：  
     
   运用了pictureBox和点击事件，当点击封面时就会自动跳转到主页面

主页：  
  
运用了button按钮和label控件，点击登录按钮可以跳转到登录界面

登录：  


登录完成后会根据登录的选择，进入不同的个人信息页面  
   
 个人详情页：  
  
  
  
  
学生和老可以在个人详情页管理自己的账号密码和个人信息，管理员可以增删改老师和学生信息。  
   
 学生和老师修改信息：  
  
  
  
老师还可以管理班级信息，删除和修改学生信息、成绩，查看班级平均分、极差、方差  
   
 管理班级页面：

如果用户使用中，输入或误操作导致程序可能奔溃的时候，会弹出提示窗口，并自动处理异常





1. 学生成绩管理系统的关键代码
   1. //用于保存和读取数据  
        
      \* 获取执行文件路径

\* string str = System.Environment.CurrentDirectory;

\*/

/\*

\* 拿到了所有学生信息S

\*/

FileInfo file = new FileInfo(@"../../Data/Student/student.txt");

sr = new StreamReader(file.FullName);

string line;

string[] student;

while ((line = sr.ReadLine()) != null)

{

student = line.Trim().Split(' ');

students.Add(student[1],new Student(student[0],student[1],int.Parse(student[2]),student[3],student[4],student[5],double.Parse(student[6]),double.Parse(student[7]),double.Parse(student[8])));

}

sr.Close();

/\*

\* 保存学生信息

\*/

FileInfo file = new FileInfo(@"../../Data/Student/student.txt");

sw = new StreamWriter(file.FullName);

foreach (var item in students)

{

sw.WriteLine(item.Value.ToString());

}

sw.Close();

* 1. 异常处理  
     try{  
      …  
     }  
     catch{  
      …  
     }  
     finally{  
      …  
     }
  2. 是否退出/保存的提示框  
     if (MessageBox.Show("你确定要退出？数据可能会丢失！", "系统提示", MessageBoxButtons.YesNo, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1) == DialogResult.Yes)

{

if (MessageBox.Show(" 是否保存？", "系统提示", MessageBoxButtons.YesNo, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1) == DialogResult.Yes)

{

data.saveData();

e.Cancel = false; //关闭窗体

}

e.Cancel = false;

}

else

{

e.Cancel = true; //返回窗体

}

* 1. 将班级数据全部输出到窗体  
       
     listBox1.Items.Clear();

groupBox1.Visible = false;

listBox1.Visible = true;

Class \_class = teacher.getClasses()[tn.Text];

double[] strs = teacher.getClasses()[tn.Text].showAverages();

listBox1.Items.Add("平均成绩：");

listBox1.Items.Add("语文：" + strs[0] + " 数学：" + strs[1] + " 英语：" + strs[2]);

listBox1.Items.Add("");

listBox1.Items.Add("方差：");

strs = teacher.getClasses()[tn.Text].showVariance();

listBox1.Items.Add("语文：" + strs[0] + " 数学：" + strs[1] + " 英语：" + strs[2]);

listBox1.Items.Add("");

strs = teacher.getClasses()[tn.Text].showRange();

listBox1.Items.Add("极差：");

listBox1.Items.Add("语文：" + strs[0] + " 数学：" + strs[1] + " 英语：" + strs[2]);

listBox1.Items.Add("");

listBox1.Items.Add("--------------学生成绩表---------------");

listBox1.Items.Add("姓名\t语文\t数学\t英语\t学号");

foreach (var studet in \_class.getAllStudents())

{

listBox1.Items.Add(studet.Value.\_ToString());

}

* 1. 防止乱输入，导致程序出错  
       
     try

{

Teacher teacher = new Teacher(ID, name, int.Parse(age), id, password);

string[] classes = \_class.Trim().Split(' ');

foreach (string cls in classes)

{

teacher.addClass(data.getClass()[cls]);

}

data.getTeachers()[name] = teacher;

cleanTe();

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

Console.WriteLine("年龄请输入纯数字！");

MessageBox.Show("年龄请输入纯数字！", "提示！！！", MessageBoxButtons.OK, MessageBoxIcon.Information);

}