

**2017 /2018 学年第 1 学期**

**本科生考查课程计算机编程提高（1）**

**综合大作业实验报告**

**（含6道编程题，与1道在线附加测试题）**

**6道编程题截止时间：2018年1月7日周日**

**1道附加题完成时间：2018年1月4日周四**

**纸质版提交截止时间：2018年1月9日周二下午17:00**

**课程名称： 计算机编程提高（1）**

**课程代码： 13003120**

**学生姓名： 齐堃**

**专 业： 信息管理与信息系统**

**学 号： 1513490426**

**学 院： 管理学院**

我想知道你是谁。设计一个窗体，窗体上面有一个显示按钮，当我点击该按钮，在窗体中，显示你的2寸照片，姓名，学号。

1. 源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

label1.Text = "1513490426" + " " + "齐堃";

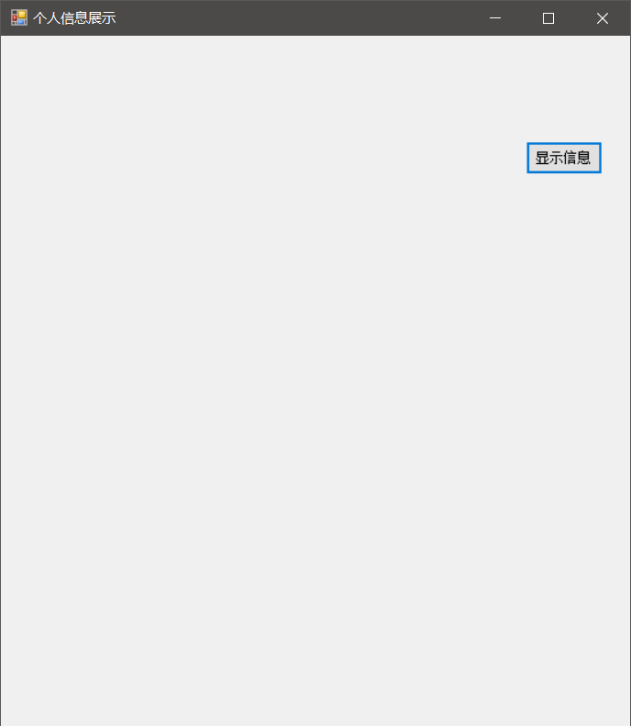
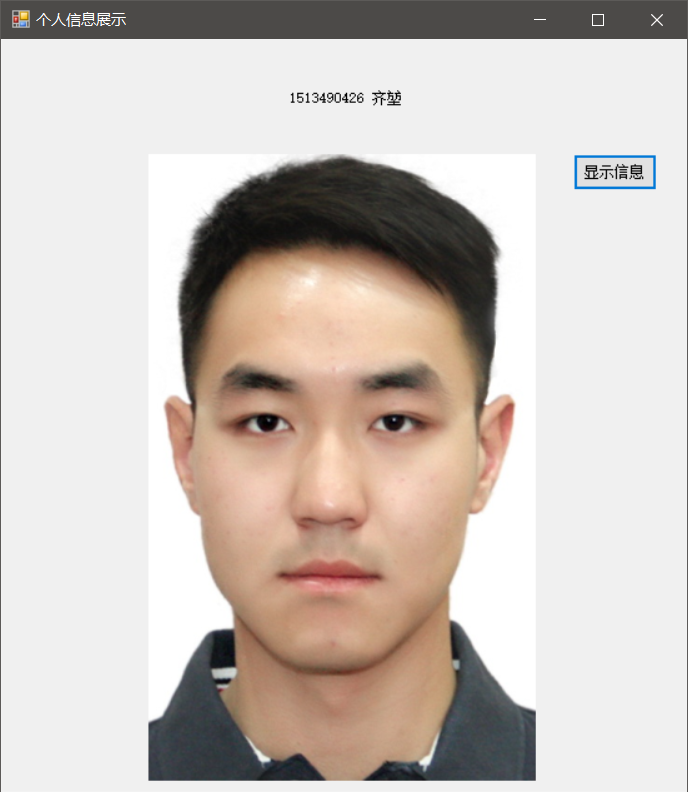
pictureBox1.ImageLocation=@"C:\Users\Issac\Documents\C#\期末\biaozhunzhao.jpg";

}

}

}

1. 运行截图：



1. 思路心得：

在窗体内设计一个pictureBox控件和一个label控件，在指定路径位置放置好2寸照片文件，然后通过一个button控件的点击事件触发：设置pictureBox的ImageLocation属性设置为照片文件路径，label的Text属性设置为“1513490426 齐堃”的文字。即可通过“显示信息”按钮实现点击该按钮显示题目要求信息。

20次猜数问题。随机生成1个1-100内的整数。在窗体中有文本框，允许用户输入整数；有猜数按钮，每猜一次，告诉用户猜大了，或者猜小了。最多只能猜20次。看用户多少次猜对。

1. 源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Random rnd = new Random();

int a = 0;

int count = 1;

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

int x = int.Parse(textBox1.Text);

if (x > a)

{

label2.Text = "猜得大了";

label3.Text = "第"+count.ToString() + "次";

++count ;

if (count == 20)

{

MessageBox.Show("猜到第20次了失败了hhh", "失败！", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Enabled = false;

button1.Enabled = false;

}

}

if (x < a)

{

label2.Text = "猜得小了";

label3.Text = "第" + count.ToString() + "次";

++count ;

if (count == 20)

{

MessageBox.Show("猜到第20次了！失败了hhh", "失败！", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Enabled = false;

button1.Enabled = false;

}

}

if (x == a)

{

label2.Text = "猜对了";

label3.Text = "第" + count.ToString() + "次";

label4.Text = "第" + count.ToString() + "次猜对";

}

}

private void Form1\_Load(object sender, EventArgs e)

{

a = rnd.Next(1, 101);

//label2.Text = a.ToString();

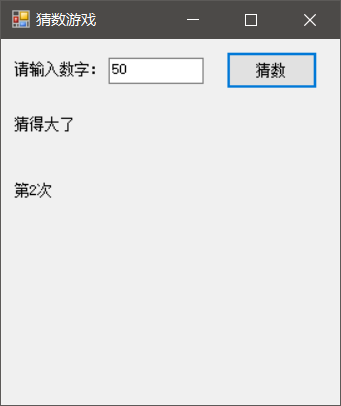
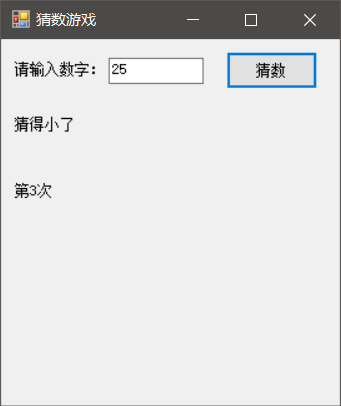
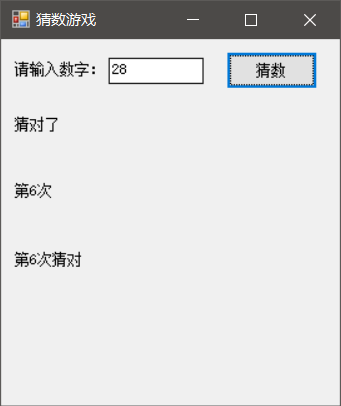
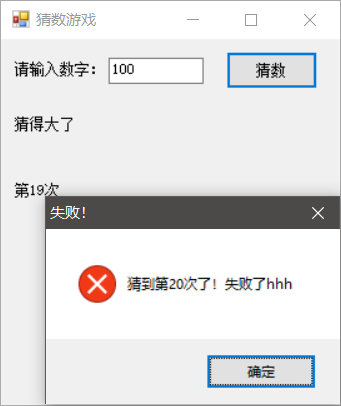
}

}

}

1. 运行截图：



1. 思路心得：

在窗体内设计一个textBox控件一个button控件，以及三个label控件负责显示相关信息。

新建随机数类的实例rnd，设置一个猜数目标值a，计数器count。均设其为全局变量。

在窗体加载段从1-100中利用rnd为a赋予一个随机值。

在button中实现：

为x赋值为textBox内的数字。接下来判断：

1. 如果x大于a，那么label2显示“猜得大了”，label3显示已经猜到了第几次，然后计数器count自增1，最后再判断：如果count已经等于20，那么弹出消息框提示已经猜到20次，说明猜数失败，将textBox和button的交互属性设置为false，禁止用户继续猜数。
2. 如果x小于a，那么label2显示“猜得小了”，label3显示已经猜到了第几次，然后计数器count自增1，最后再判断：如果count已经等于20，那么弹出消息框提示已经猜到20次，说明猜数失败，将textBox和button的交互属性设置为false，禁止用户继续猜数。
3. 如果x等于a，那么在label2显示“猜对了”，label3显示已经猜到了第几次，label4中显示是第几次猜对的。

在窗体中能够输入日期，然后点击按钮，在窗体中在显示这一天是这一年的第几天，属于这一年的第几周。

1. 源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Globalization;

namespace q3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

DateTime dt1 = Convert.ToDateTime(textBox1.Text);

GregorianCalendar haha = new GregorianCalendar();

int weekofyear = haha.GetWeekOfYear(dt1, CalendarWeekRule.FirstDay, DayOfWeek.Monday);

label1.Text = "这天是这年的第" + dt1.DayOfYear.ToString() + "天";

label2.Text = "这天所在周属于该年的第" + weekofyear.ToString() + "周";

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

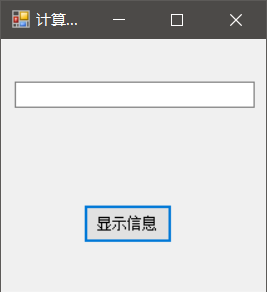
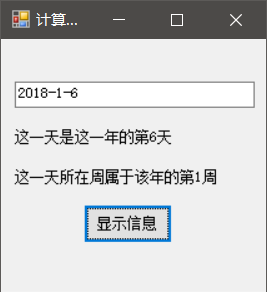
{

}

}

}

1. 运行截图：

1. 思路心得：

在窗体中设计一个textBox控件，一个button控件，以及两个label控件。

在button控件内写入：

新建DateTime类实例dt1并将textBox内的日期转换赋值给它，新建一个GregorianCalendar类实例haha。

通过实例haha的方法GetWeekOfYear获取该天所在周是该年的第几周并赋值给变量weekofyear

在label1显示该天是这一年的第几天，在label2显示该天所在周是该年的第几周。

我想知道你的2寸照片的负片（负片是经曝光和显影加工后得到的影像，其明暗与被摄体相反，其色彩则为被摄体的补色，它需经印放在照片上才还原为正像），点击按钮实现。

1. 源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

int height = pictureBox1.Image.Height;

int width = pictureBox1.Image.Width;

Bitmap newbitmap = new Bitmap(width, height);

Bitmap oldbitmap = (Bitmap)pictureBox1.Image;

Color pixel;

for (int x = 1; x < width; x++)

{

for (int y = 1; y < height; y++)

{

int r, g, b;

pixel = oldbitmap.GetPixel(x, y);

r = 255 - pixel.R;

g = 255 - pixel.G;

b = 255 - pixel.B;

newbitmap.SetPixel(x, y, Color.FromArgb(r, g, b));

}

}

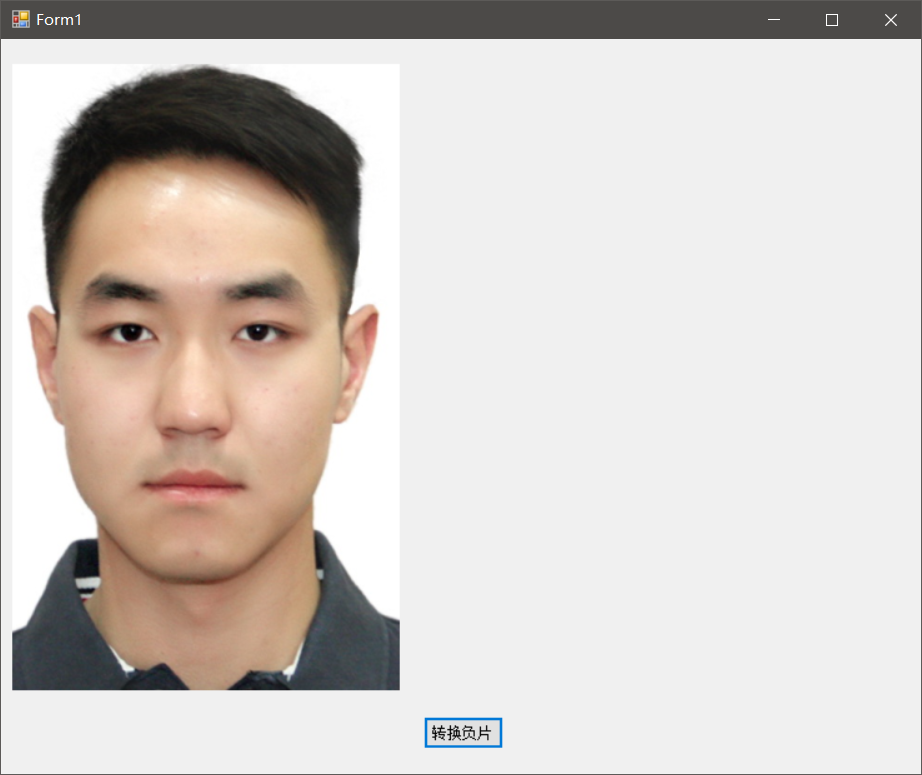
pictureBox2.Image = newbitmap;

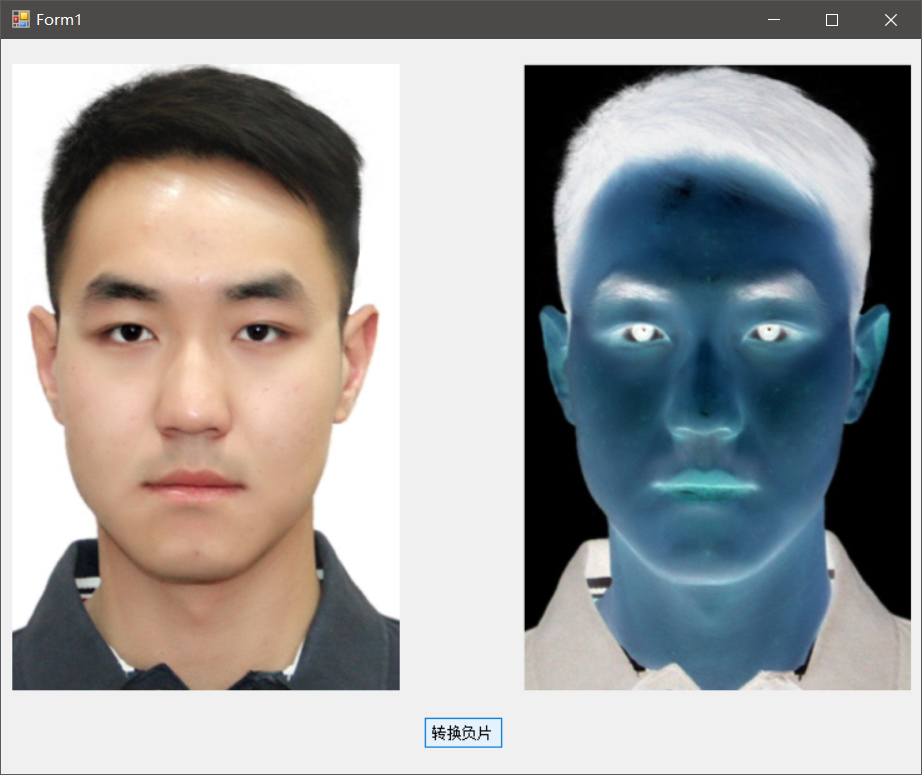
}

}

}

1. 运行截图：





1. 思路心得：

在窗体上设计两个pictureBox控件和一个button控件，为第一个pictureBox1控件导入2寸照片文件。

在button中写入：

获取2寸照片文件的高度和宽度，以获取到的高宽新建一个Bitmap实例newbitmap，再将2寸照片位图化新建Bitmap实例oldbitmap，在设定一个颜色实例pixel。

然后嵌套循环实现：

将每个像素的颜色值反转（以255-像素的每种颜色），然后再将新的像素写回新的newbitmap中。

最后将转换完成的newbitmap放入pictureBox2中显示。

编写一个C#程序，设置用于演讲比赛的时间列表。输入文本框中填：起始时间，结束时间，每组演讲时间。然后点击按钮，将排序号的时间，输出到列表框。例如：输入起始时间9：00，结束时间10:00，每组演讲时间15分钟。请输出：

9:00-9:15

9:15-9:30

9:30-9:45

9:45-10:00

1. 源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

DateTime dt1 = Convert.ToDateTime(textBox1.Text);

DateTime dt2 = Convert.ToDateTime(textBox2.Text);

int span = int.Parse(textBox3.Text);

DateTime dt3 = new DateTime();

TimeSpan ts1 = dt2.Subtract(dt1);

int sign = 0;

if ((ts1.Minutes + ts1.Hours \* 60) % span != 0)

{

MessageBox.Show("时间跨度不是"+span.ToString()+"的倍数，最后一个小组时间不足！", "时间跨度有误！", MessageBoxButtons.OK, MessageBoxIcon.Error);

sign = 1;

}

if (dt1 == dt2)

{

MessageBox.Show("起始与结束时间不能相同！", "输入时间有误！", MessageBoxButtons.OK, MessageBoxIcon.Error);

sign = 1;

}

while (dt3 != dt2 && sign == 0)

{

dt3 = dt1.AddMinutes(span);

listBox1.Items.Add(dt1.ToString("hh:mm") + "-" + dt3.ToString("hh:mm") + "\n");

dt1 = dt3;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

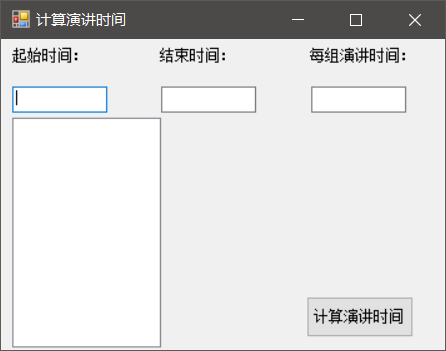
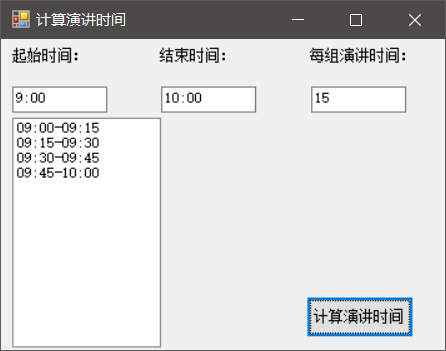
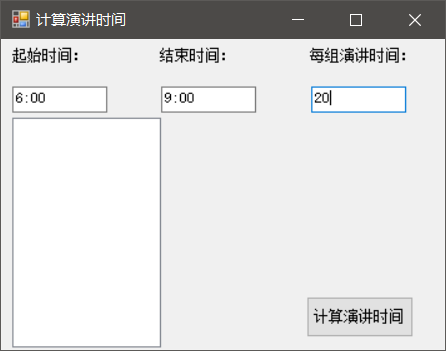
{

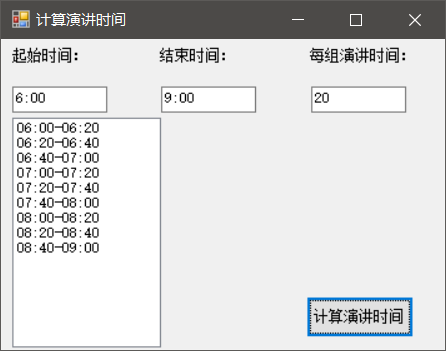
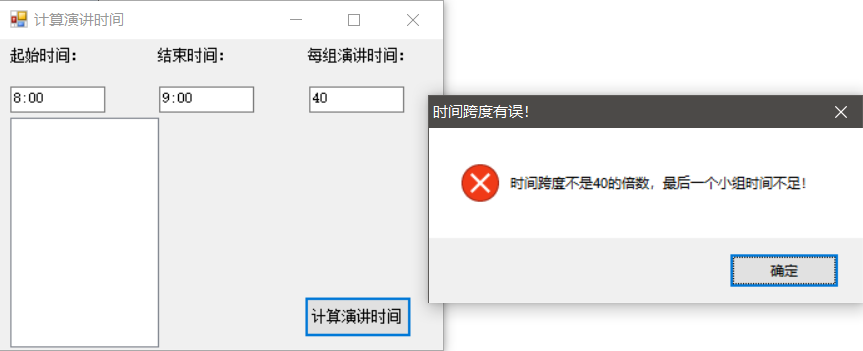
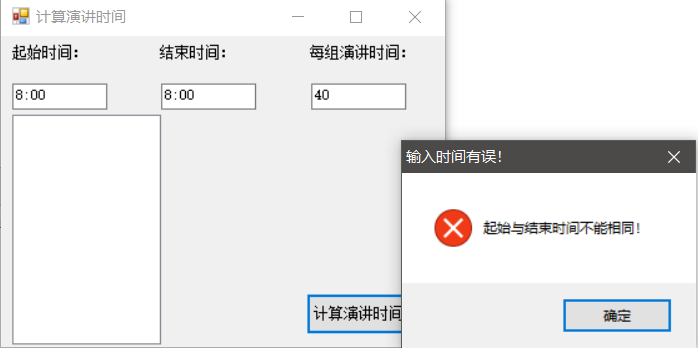
}

}

}

1. 运行截图：





1. 思路心得：

在窗体上设计三个textBox控件分别负责接收起始时间、结束时间、每组演讲时间，一个listBox控件以及一个button控件

在button控件内写入：

新建DateTime实例dt1并将textBox1的起始时间值转换并赋给它，新建DtaeTime实例dt2并将textBox2的结束时间值转换并赋给它，由变量span来接收设定的每组演讲时间，再新建一个DateTime实例dt3，新建TimeSpan实例ts1并将dt2与dt1之间的时间跨度值赋给它，设定一个信号量值sign并设其初始值是0。

接下来判断：

1. 如果时间跨度ts1并不是每组演讲时间span的整倍数，那么就弹出消息框提示时间跨度并不是每组演讲时间的整倍数，最后一组的时间将不足，并将信号量值sign置1。
2. 如果起始时间dt1等于结束时间dt2，那么就弹出消息框提示起始时间不能与结束时间相同，并将信号量置1。

当dt3不等于结束时间dt2时且信号量sign值为0，那么循环：

dt3等于dt1每次增加与每组演讲时间span等量的分钟数，然后在listBox的项目中增加dt1的值到dt3的值表示该小组演讲时间的始末，最后再将dt3的值赋给dt1。

如此即可循环输出每组演讲的起始与终止时间，信号量的作用在于确保输入信息的正确性，如果sign值被置为1那么代表时间有误，那么在消息框提示过后循环也不会进行，以避免循环在无控制条件下运行导致进入死循环；如果sign为0那么说明时间无误，循环可以正常进行，即正确输出结果。

微型学生信息管理系统。要求在窗体中设计菜单，以及学生信息表的管理，仅需次1张表的处理。通过菜单选择增删改学生信息。要求设计学号、姓名，性别，出生年月，籍贯，专业，手机号，Email 地址，要求添加好10个同学，含自己的信息，自己学号的尾号是数字几，则在这10个学生中排序第几。如尾号是9，则在10个学生中排序9。报告中含学生信息的截图，数据库的截图。

1. 源代码：

**主界面（Form1.cs）：**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

namespace q6

{

public partial class Form1 : Form

{

public Form1()

{

pForm1 = this;

InitializeComponent();

}

public static Form1 pForm1 = null;

OleDbDataAdapter adapter;

public DataTable table = new DataTable();

string str = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=student.mdb";

private void Form1\_Load(object sender, EventArgs e)

{

}

private void 查看数据集ToolStripMenuItem\_Click(object sender, EventArgs e)

{

table.Clear();

string sql = "Select \* from student";

adapter = new OleDbDataAdapter(sql, str);

OleDbCommandBuilder builder = new OleDbCommandBuilder(adapter);

adapter.DeleteCommand = builder.GetDeleteCommand();

adapter.InsertCommand = builder.GetInsertCommand();

adapter.UpdateCommand = builder.GetUpdateCommand();

adapter.Fill(table);

table.DefaultView.Sort = "学号 asc";

dataGridView1.DataSource = table;

}

private void 添加记录ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form2 add = new Form2();

add.ShowDialog();

DataRow row = table.NewRow();

row["学号"] = add.num;

row["姓名"] = add.name;

row["性别"] = add.sex;

row["出生年月"] = add.birth;

row["籍贯"] = add.home;

row["专业"] = add.major;

row["手机号"] = add.phnum;

row["Email地址"] = add.email;

table.Rows.Add(row);

dataGridView1.EndEdit();

adapter.Update(table);

MessageBox.Show("已成功添加记录！", "添加学生信息", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

private void 修改记录ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form3 update = new Form3();

update.ShowDialog();

OleDbConnection conn = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

conn.ConnectionString = str;

conn.Open();

cmd.Connection = conn;

cmd.CommandText = "Update student Set 姓名='" + update.name + "',性别='" + update.sex + "',出生年月='" + update.birth + "',籍贯='" + update.home + "',专业='" + update.major + "',手机号='" + update.phnum + "',Email地址='" + update.email + "' Where 学号='"+update.num+"'";

int val = cmd.ExecuteNonQuery();

if (val > 0)

MessageBox.Show("已成功修改指定记录！", "修改指定记录", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

else

MessageBox.Show("未找到目标记录，请确认信息！", "修改指定记录", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

private void 删除记录ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form4 delete = new Form4();

delete.ShowDialog();

OleDbConnection conn = new OleDbConnection();

OleDbCommand cmd = new OleDbCommand();

conn.ConnectionString = str;

conn.Open();

cmd.Connection = conn;

cmd.CommandText = "Delete From student Where 学号='" + delete.num + "'";

int val = cmd.ExecuteNonQuery();

if (val > 0)

MessageBox.Show("已成功删除指定记录！", "删除指定记录", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

else

MessageBox.Show("未找到目标记录，请确认信息！", "删除指定记录", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}

**添加界面（Form2.cs）：**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q6

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public int num = 0;

public string name = "";

public string sex = "";

public string birth = "";

public string home = "";

public string major = "";

public string phnum = "";

public string email = "";

private void Form2\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Form1.pForm1.Show();

num = int.Parse(textBox1.Text);

name = textBox2.Text;

sex = textBox3.Text;

birth = textBox4.Text;

home = textBox5.Text;

major = textBox6.Text;

phnum = textBox7.Text;

email = textBox8.Text;

this.Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

}

}

}

**修改界面（Form3.cs）：**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q6

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

public int num = 0;

public string name = "";

public string sex = "";

public string birth = "";

public string home = "";

public string major = "";

public string phnum = "";

public string email = "";

private void Form3\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Form1.pForm1.Show();

num = int.Parse(textBox1.Text);

name = textBox2.Text;

sex = textBox3.Text;

birth = textBox4.Text;

home = textBox5.Text;

major = textBox6.Text;

phnum = textBox7.Text;

email = textBox8.Text;

this.Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

}

}

}

**删除界面（Form4.cs）：**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace q6

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

public string num = "";

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Form1.pForm1.Show();

num = textBox1.Text;

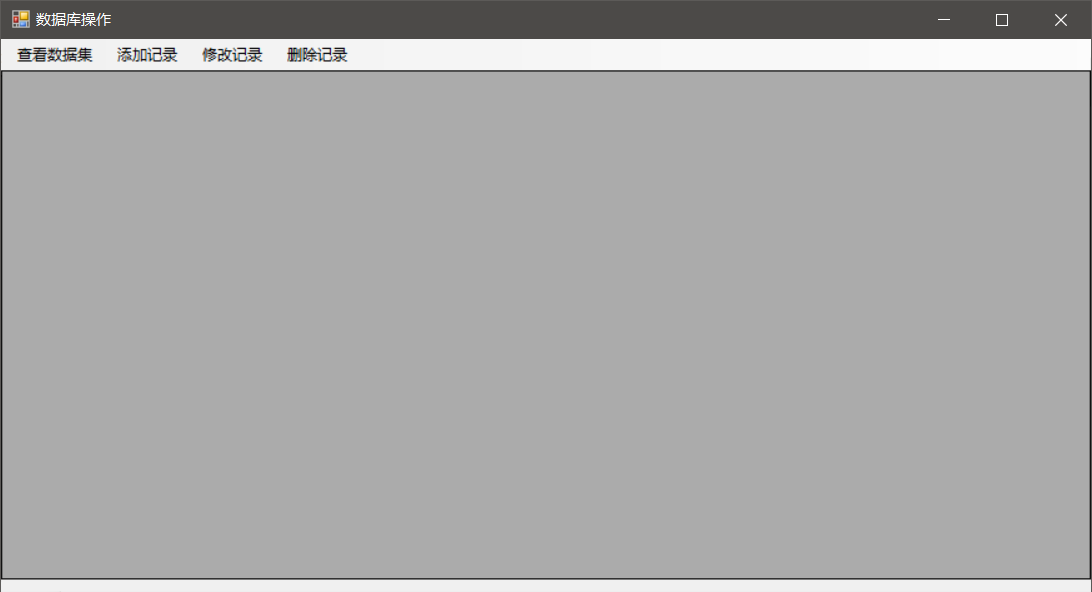
this.Close();

}

}

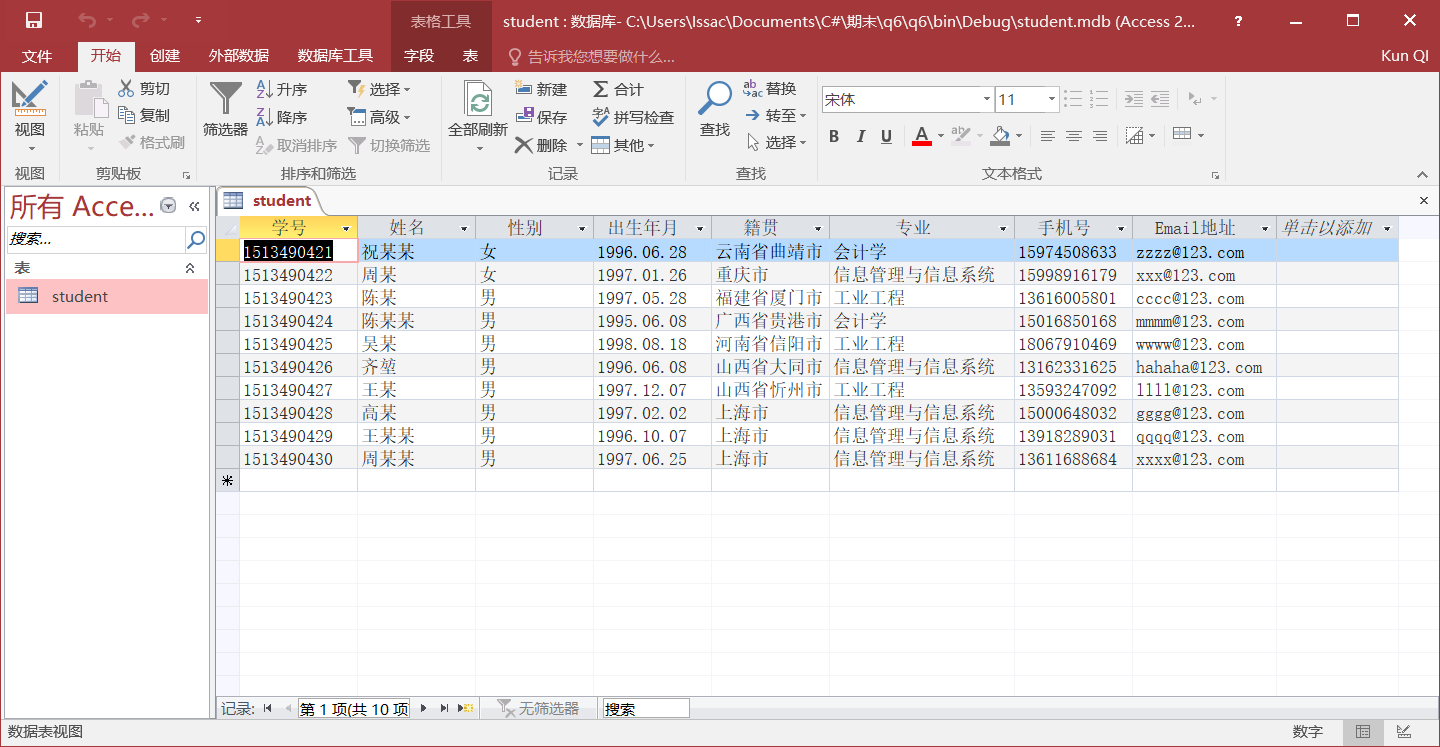
}

1. 运行截图：



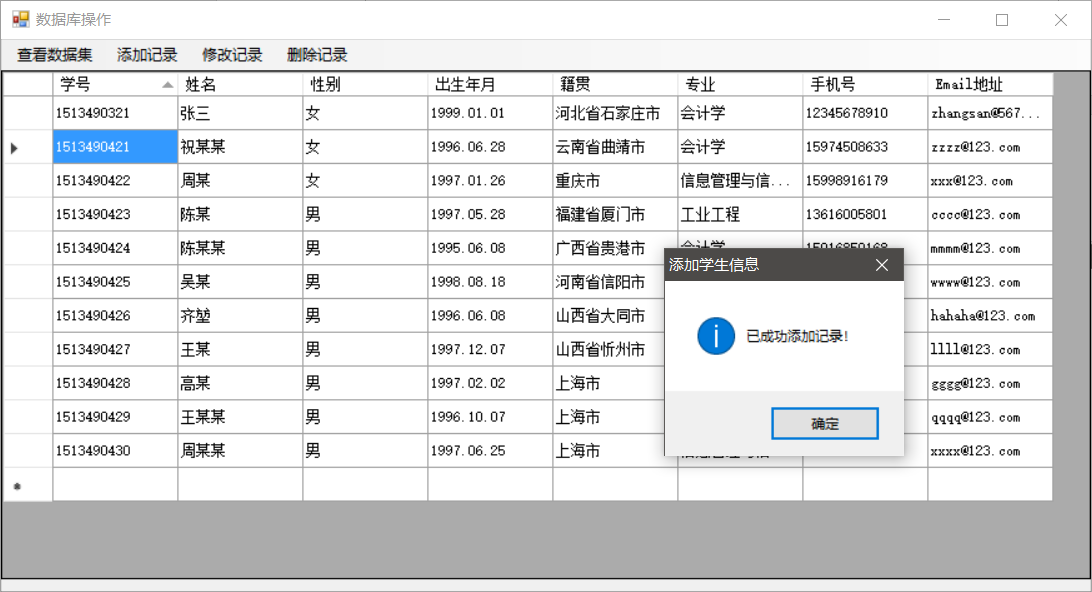
**学生信息：**





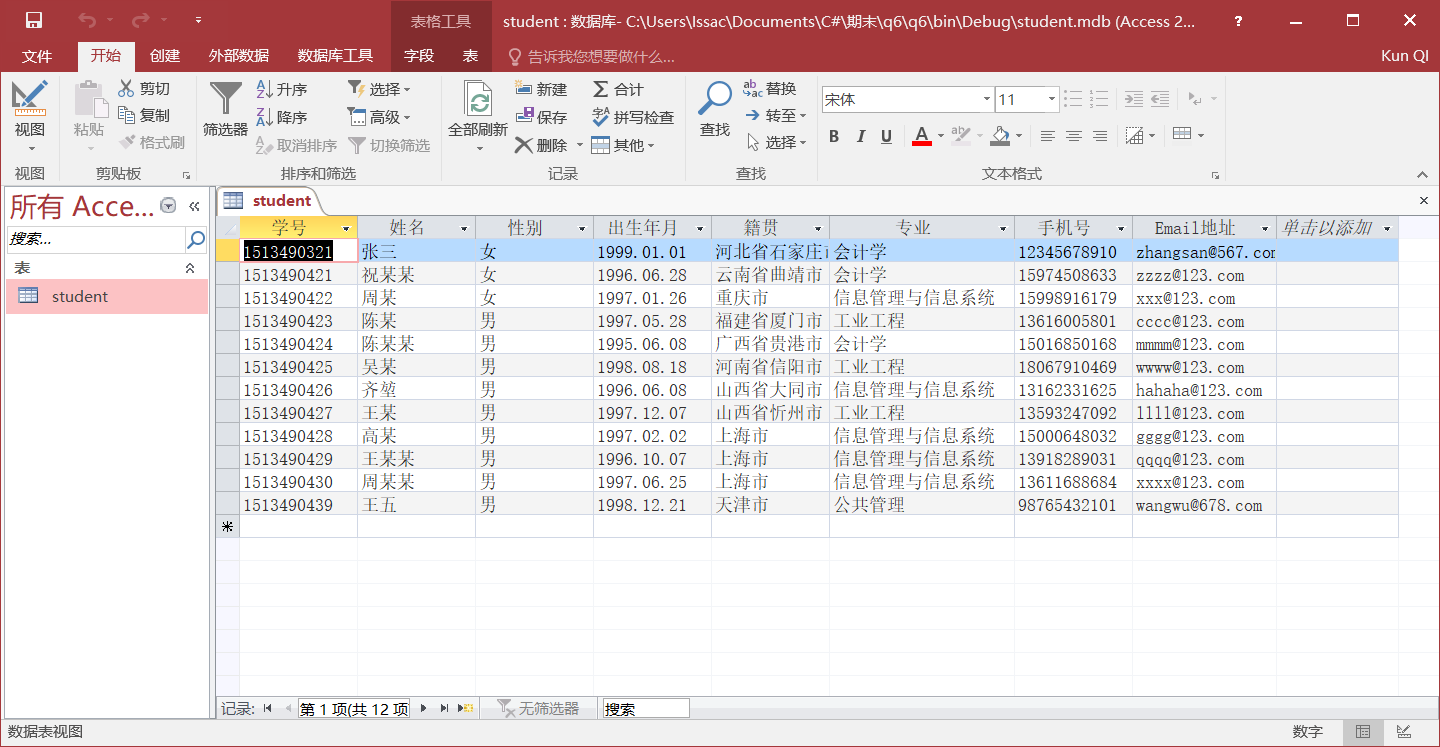
**添加信息：**









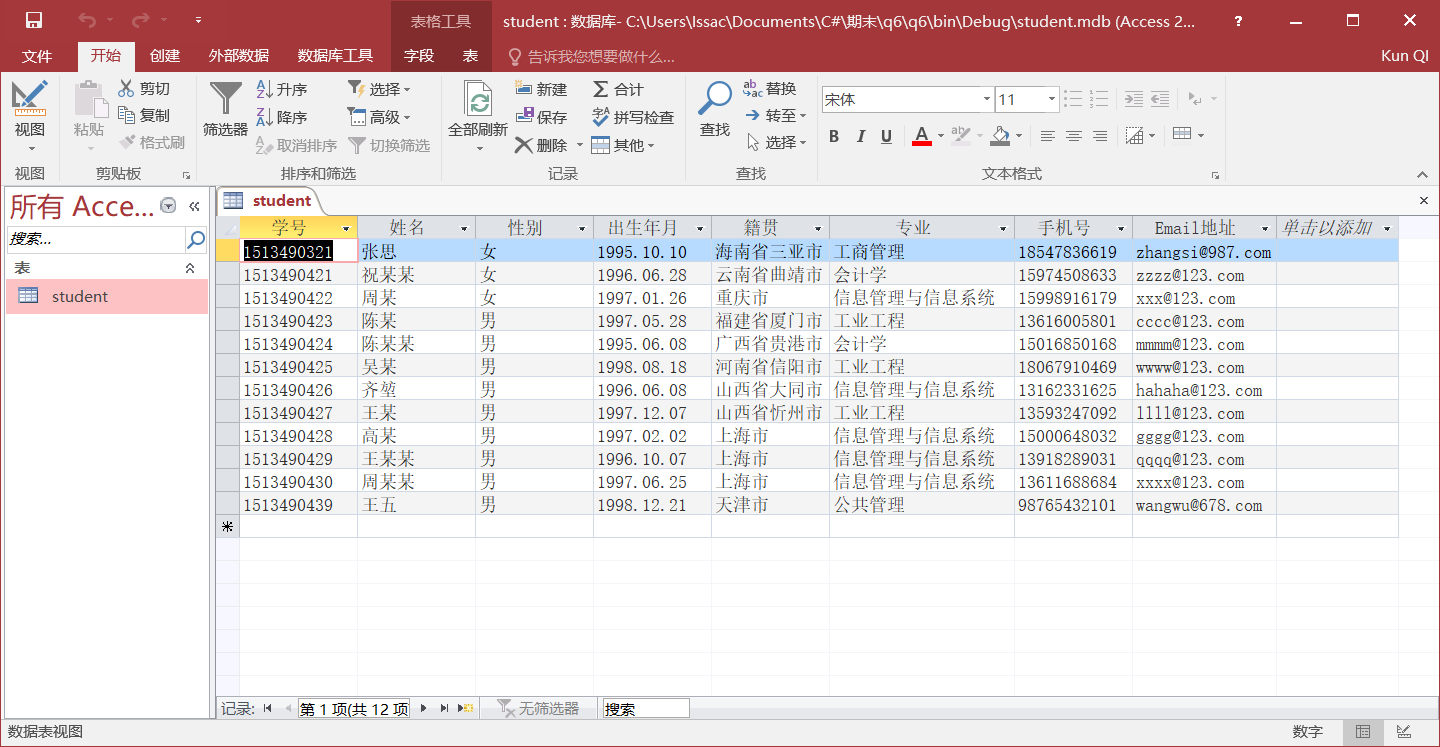


**修改信息：**







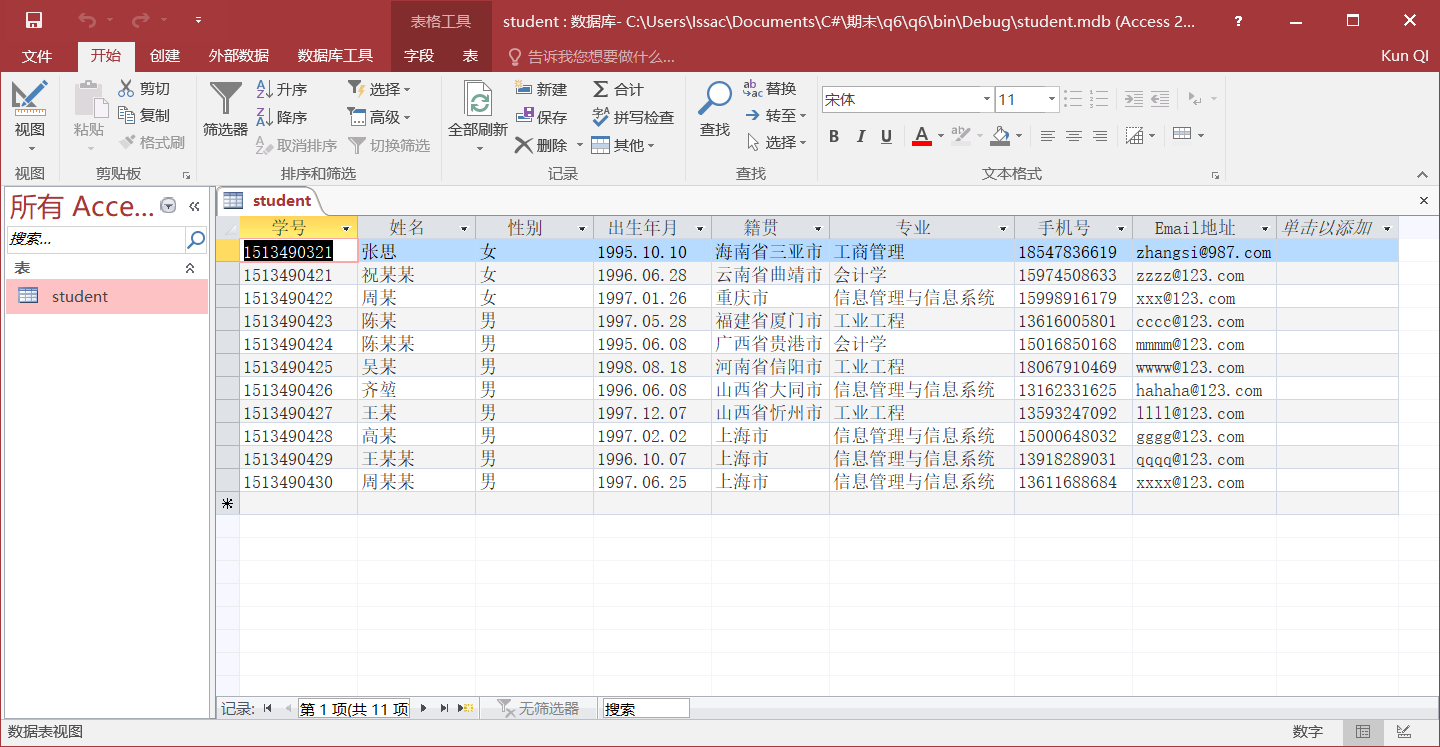


**删除信息：**









1. 思路心得：

在主界面窗体内设计菜单——包含4个选项：查看数据集、添加记录、修改记录以及删除记录。单击“查看数据集”可以使dataGridView控件连接并显示数据库信息，单击“添加记录”会弹出添加记录的窗体接收信息的添加，单击“修改记录”会弹出修改记录的窗体用于接收需要修改的信息，单击“删除记录”会弹出接收删除目标学号的窗体用于删除指定记录。

首先在主界面Form1中设置Form1的实例静态变量pForm1，用于其他窗口调回主界面窗口。再创建DataAdapter实例adapter和DataTable实例table，书写连接数据库的语句。

接下来分别在菜单各按钮中实现数据库操作：

1. 查看数据集：

首先将table清空，然后书写数据库查询语句，由adapter连接数据库并执行查询语句获得数据，由OleDbCommandBuilder实例builder获取数据库操作方法，用adapter的Fill方法将数据填充到table中，并使其保持升序排列，设置dataGridView的数据源为table，即可将数据库数据显示在dataGridView控件内。实现查看数据库操作。

1. 添加记录：

首先新建添加记录窗体Form2的实例add，使用add的方法ShowDialog调出Form2并使Form1挂起。在Form2中设计对应数据库信息的各属性和文本框来接收数据，在button控件中将数据赋给已经创建的对应的全局变量供Form1调用，并关闭该Form2窗体。在调回Form1后，Form1创建DataRow的实例row，调用Form2的全局变量将值添加到列当中，再将当前行添加到DataTable中，关闭dataGridView的修改，最后把更新后的数据集保存到数据库。实现添加信息并弹出消息框通知用户。

1. 修改记录：

首先创建修改记录窗体Form3的实例update，使用update的方法ShowDialog调出Form3并使Form1挂起。在Form3中设计对应数据库信息的各属性和文本框来接收数据，在button控件中将数据赋给已经创建的对应的全局变量供Form1调用，并关闭Form3窗体。调回Form1窗体后，创建OleDbConnection的实例conn和OleDbCommand的实例cmd，通过conn连接数据库并打开，利用cmdCommandText方法书写调用Form3的全局变量并输入操作语句通知数据库修改数据的语句，用变量val捕捉cmd的ExecuteNonQuery方法执行的修改结果，并根据捕捉到的信息弹出相应的消息框提示用户。

1. 删除记录：

首先创建删除记录窗体Form4的实例delete，使用delete的方法ShowDialog调出Form4并使Form1挂起。在Form4中设计文本框接收需要删除的目标数据的学号，在button控件中将数据赋给已经创建的全局变量供Form1调用，并关闭Form4窗体。调回Form1窗体后，通过conn连接数据库并打开，利用cmdCommandText方法书写调用Form4的全局变量并输入操作语句通知数据库删除目标数据的语句，用变量val捕捉cmd的ExecuteNonQuery方法执行的删除结果，并根据捕捉到的信息弹出相应的消息框提示用户。

额外题.

设计一个窗体程序，具备菜单栏，可以打开文本文件，并且在窗体中读取文本文件内容，能够分别计数文本文件内容的中文字符数、英文字符数、阿拉伯数字数、其他字符数、以及文本的行数并在消息框中显示结果，还可以点击按钮结束程序，另外要在下方状态栏显示自己的学生信息。

1. 源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace EWai

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void 打开文件ToolStripMenuItem\_Click(object sender, EventArgs e)

{

DialogResult myresult = openFileDialog1.ShowDialog();//打开对话框

if (myresult == DialogResult.OK)

{

StreamReader sr = new StreamReader(openFileDialog1.FileName, Encoding.Default);//创建字节流

richTextBox1.Text = sr.ReadToEnd();//以字节流形式将文本文件读入富文本框

}

}

private void Form1\_Load(object sender, EventArgs e)

{

openFileDialog1.Filter = "文本文件|\*.txt";//设置文件过滤

toolStripStatusLabel1.Text = "学号：1513490426 姓名：齐堃";//制作者信息

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void toolStripStatusLabel1\_Click(object sender, EventArgs e)

{

}

private void 统计字数ToolStripMenuItem\_Click(object sender, EventArgs e)

{

int chnnum = 0;//初始化中文字符数

int engnum = 0;//初始化英文字符数

int nbrnum = 0;//初始化阿拉伯数字数

int bionum = 0;//初始化其他字符数

int linnum = 1;//初始化文本行数

foreach (char ch in richTextBox1.Text)

{

if (ch >= 0x4e00 && ch <= 0x9fbb)//如果为中文字符，相应计数器递增

chnnum++;

if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))//如果为英文字符，相应计数器递增

engnum++;

if (ch >= '0' && ch <= '9')//如果为数字，相应计数器递增

nbrnum++;

if ("`~!@#$%^&\*()\_+-={}[]:\";'<>,.?/\\| ".IndexOf(ch) != -1)//如果为其他字符，相应计数器递增

bionum++;

if (ch == '\n')//如果有换行标识，相应计数器递增

linnum++;

}

MessageBox.Show(string.Format("中文字数为：{0}\n英文字数为：{1}\n数字字数为：{2}\n其他字符数为：{3}\n行数为：{4}\n", chnnum, engnum, nbrnum, bionum, linnum), "统计字数", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);//在消息框显示结果

}

private void 结束运行ToolStripMenuItem\_Click(object sender, EventArgs e)

{

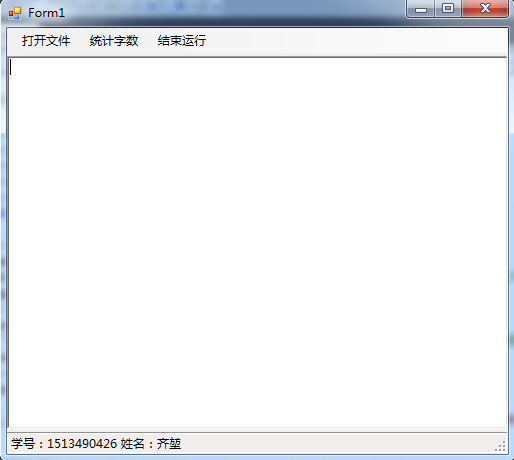
Application.Exit();//关闭程序

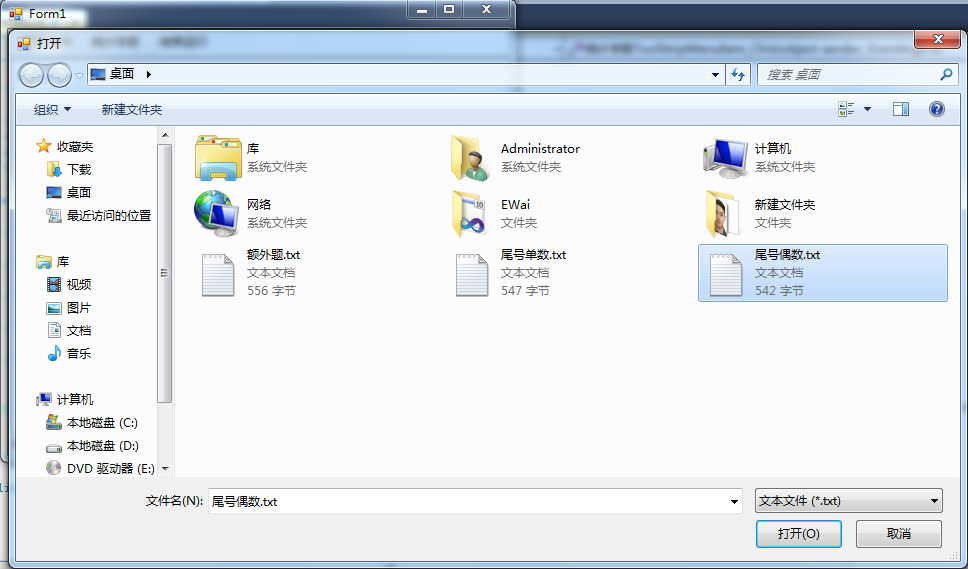
}

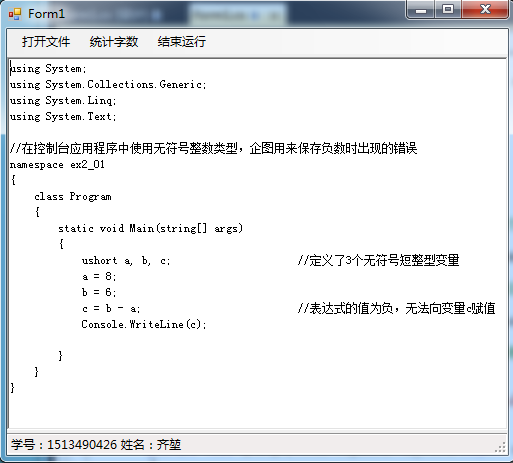
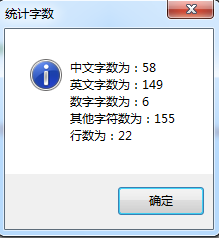
}

}

1. 运行截图：







1. 思路心得：

在窗体上设计菜单实现以下功能：

1. 打开文件：在窗体加载时以Filter方法设置文件过滤为.txt格式文件。新建DialogResult实例myresult来打开对话框，如果打开对话框成功，则将文本文件以字符流的形式读入富文本框。
2. 统计字数：分别为需要统计的各项字符以及行数设置计数器，并初始化。利用foreach遍历富文本框中的字符：分别利用if语句设计各类字符的判断条件，当字符符合某种判断条件时，相应字符的计数器自增1。当遍历完成后，弹出消息框显示各计数器统计结果。
3. 关闭程序：利用Application的Exit方法实现程序的关闭。
4. 设计状态栏，并在窗体加载时设置状态栏显示“学号：1513490426 姓名：齐堃”的文本。