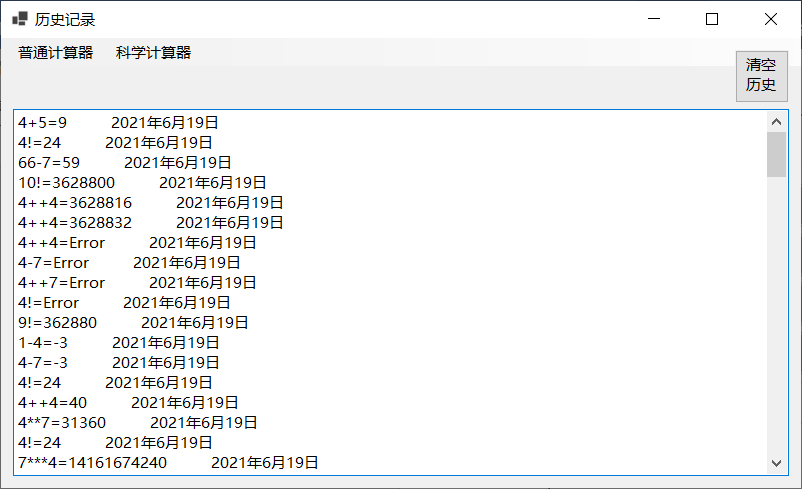
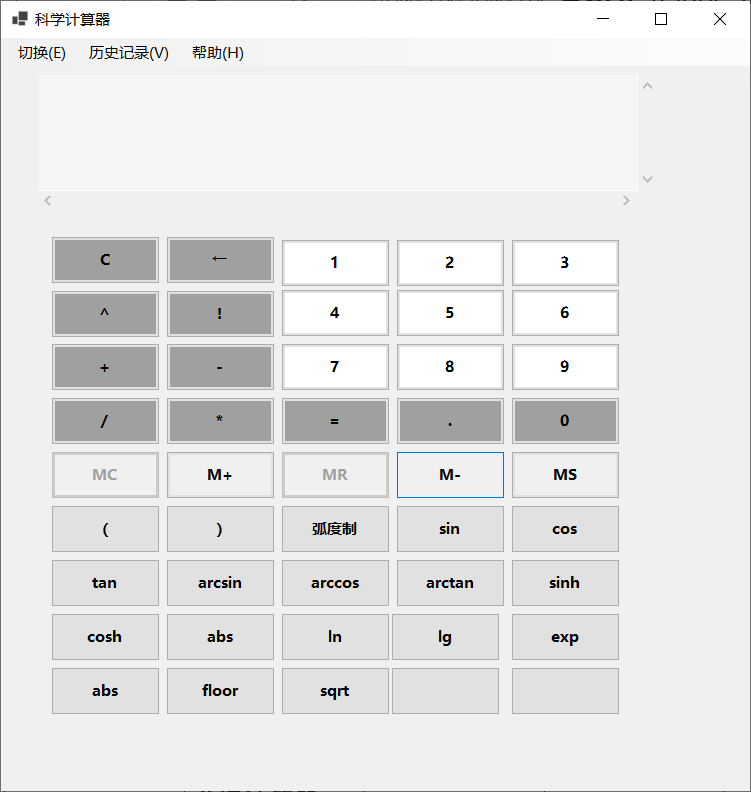
# 科学计算器代码及界面设计

## 界面设计







## 属性界面

**Form1窗口属性事件一览**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **text** | **size** | **load** |
| **Form1** | **普通计算器** | **767，600** | **Form1\_Load** |

**Form1中button按钮属性事件一览**

|  |  |  |
| --- | --- | --- |
| **Name** | **Text** | **Click** |
| **button1** | **1** | **button1\_Click** |
| **button2** | **2** | **button1\_Click** |
| **button3** | **3** | **button1\_Click** |
| **button4** | **4** | **button1\_Click** |
| **button5** | **5** | **button1\_Click** |
| **button6** | **6** | **button1\_Click** |
| **button7** | **7** | **button1\_Click** |
| **button8** | **8** | **button1\_Click** |
| **button9** | **9** | **button1\_Click** |
| **button10** | **.** | **button1\_Click** |
| **button11** | **0** | **button1\_Click** |
| **button12** | **+** | **button1\_Click** |
| **button13** | **-** | **button1\_Click** |
| **button14** | **\*** | **button1\_Click** |
| **button15** | **/** | **button1\_Click** |
| **button16** | **=** | **button16\_Click** |
| **button17** | **^** | **button1\_Click** |
| **button18** | **!** | **button1\_Click** |
| **button19** | **←** | **button19\_Click** |
| **button20** | **C** | **button20\_Click** |
| **button21** | **MC** | **button21\_Click** |
| **button22** | **M+** | **button22\_Click** |
| **button23** | **MR** | **button23\_Click** |
| **button24** | **M-** | **button24\_Click** |
| **button25** | **MS** | **button25\_Click** |

**Form1中textbox属性事件一览**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **size** | **Scrollbar** | **Multiline** |
| **textbox1** | **620,173** | **Both** | **Ture** |

**Form1中meunStrips属性事件一览**

|  |  |  |
| --- | --- | --- |
| **Name** | **Text** | **Click** |
| **切换ToolStripMenuItem** | **切换(&E)** | **切换ToolStripMenuItem\_Click** |
| **历史记录ToolStripMenuItem** | **切换(&V)** | **历史记录ToolStripMenuItem\_Click** |
| **帮助ToolStripMenuItem** | **切换(&H)** | **帮助ToolStripMenuItem\_Click** |

**由于Form2继承Form1窗体，重复属性便不在一一重复了。**

**Form2窗口属性事件一览2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **text** | **size** | **load** |
| **Form2** | **科学计算器** | **767，800** | **Form2\_Load** |

**Form1中button按钮属性事件一览**

|  |  |  |
| --- | --- | --- |
| **Name** | **Text** | **Click** |
| **button26** | **(** | **button26\_Click** |
| **button27** | **)** | **button26\_Click** |
| **button28** | **弧度制** | **button28\_Click** |
| **button29** | **sin** | **button26\_Click** |
| **button30** | **cos** | **button26\_Click** |
| **button31** | **tan** | **button26\_Click** |
| **button32** | **arcsin** | **button26\_Click** |
| **button33** | **arccos** | **button26\_Click** |
| **button34** | **arctan** | **button26\_Click** |
| **button35** | **sinh** | **button26\_Click** |
| **button36** | **cosh** | **button26\_Click** |
| **button37** | **abs** | **button26\_Click** |
| **button38** | **ln** | **button26\_Click** |
| **button39** | **lg** | **button26\_Click** |
| **button40** | **exp** | **button26\_Click** |
| **button41** | **abs** | **button26\_Click** |
| **button42** | **floor** | **button26\_Click** |
| **button43** | **sqrt** | **button26\_Click** |

**history窗口属性事件一览2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **text** | **size** | **load** |
| **history** | **历史记录** | **818，497** | **historty\_Load** |

## 程序代码

**Form1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.Threading.Tasks;

using System.Windows.Forms;

namespace 科学计算器

{

public partial class Form1 : Form //程序自带

{

public Form1()

{

InitializeComponent();

}

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

{

Button a = (Button)sender; //sender表示当前执行事件的对象，将它由object转为Button类，并用Button a来表示。

textBox1.Text += a.Text; //由于+-\*/及0到9等一系列按钮的事件的执行内容相同，为了代码简洁，就不再一一创建事件了，直接在属性窗口事件界面引用便行了。

}

private void button16\_Click(object sender, EventArgs e)

{

eqaul(); //执行‘=’按钮的内容。

}

public virtual void eqaul() //定义一个虚方法，利于派生类重新改写。（Form2用到）

{

string s = textBox1.Lines[textBox1.Lines.Length - 1];

textBox1.Text += "=\r\n" + CC.Calculate(s);

Properties.Settings.Default.jilu += s + "=" + CC.Calculate(s) + " " + DateTime.Now.ToLongDateString() + "\r\n";

Properties.Settings.Default.Save(); //用同步属性来保存计算器的运算记录

// textBox1.Lines将文本内容中每一段文字（硬回车换行的内容）以[数组](https://so.csdn.net/so/search?q=%E6%95%B0%E7%BB%84&spm=1001.2101.3001.7020)的形式保存在Lines中 \r\n 回车、换行

}

private void Form1\_Load(object sender, EventArgs e)//用户加载窗体时发生变化

{

textBox1.Focus(); //使光标落在textBox1的开头处。

textBox1.Select(1, 0); //将光标置于TextBox 控件的内容的末尾，应调用 Select 方法，并指定选择内容的起始位置等于文本内容的长度，选择长度为0。

}

private void 切换ToolStripMenuItem\_Click(object sender, EventArgs e)

{

reswitch(); //进行窗口的切换

}

public virtual void reswitch() //虚方法（Form2中重写）

{

Form2 a = new Form2();

a.Show(); //显示Form2

this.Hide(); //隐藏Form1

}

private void button19\_Click(object sender, EventArgs e)

{

if(textBox1.Text!="")

{

string s = textBox1.Text; //当文本框不为空时，←按钮发生作用，清除文本框最后一个字符。

textBox1.Text = s.Substring(0, s.Length - 1); //截取字符串

}

}

private void button20\_Click(object sender, EventArgs e)

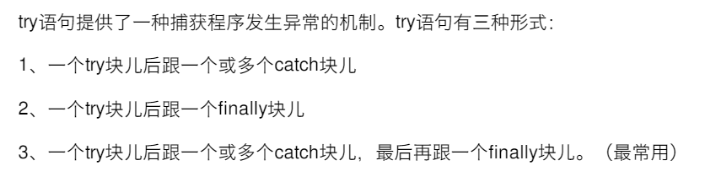
{

textBox1.Text = ""; //C按钮发生作用清空文本框

}

}

}

****

**CC代码：**

using System;

using System.Collections.Generic;

using System.Text;

namespace 科学计算器

{

class CC //普通计算类

{

public static string Calculate(string str) //采用后缀表达法来计算连续表达式

{

Stack<double> m = new Stack<double>();//数字栈

Stack<char> s = new Stack<char>();//符号栈

char[] inputStr = new char[1000]; //字符数组

try

{

double a = 0;

if (double.TryParse(str, out a)) //若输入表达式为数字，则直接返回该数 int.TryParse(n1.Text, out P\_int\_Number) 其中第⼀个参数代表被转换的参数，第⼆个参数为转换后的参数 int类型，成功返回True，失败返回False。

{

}

else

{

str = str.Replace("!", "!0"); //将!转为!0,0不起作用，只是防止破坏表达式结构，即4!0=4!;

inputStr = str.ToCharArray(); //将字符串转为字符数组

double n1, n2, sum; //n1,n2来贮存要进行计算的两个数，sum来贮存运算结果.

string t = null; //定义一个空字符串，该字符串用于将数字字符转为double型

char c1; //c1用于贮存运算符号

for (int i = 0; i < inputStr.Length; i++)

{

if (i == 0&&inputStr[i]=='-') //若第一个字符为‘-’，应判断为负号，而不是减号。

{

t += "-";

i++;

}

if (inputStr[i] >= 48 && inputStr[i] <= 57 || inputStr[i] == '.') //判断是否为数字或小数点

{

while (i < inputStr.Length && ((inputStr[i] >= 48 && inputStr[i] <= 57) || inputStr[i] == '.'))

{

t += inputStr[i]; //将运算符之间的字符转为数并存入数字栈

i++;

}

i--;

double num = Convert.ToDouble(t);

t = "";

m.Push(num);

}

else if (inputStr[i] == '+' || inputStr[i] == '\*' || inputStr[i] == '-' || inputStr[i] == '/' || inputStr[i] == '^' || inputStr[i] == '!') //判断是否为加减乘除

{

if (s.Count.Equals(0)) //若符号栈为空，直接丢入运算符

{

s.Push(inputStr[i]);

}

else if (OperatorPrecedence(inputStr[i]) > OperatorPrecedence(s.Peek()))

{

s.Push(inputStr[i]); //若当前运算符比前一个运算符优先级高，则直接丢入，不弹出前一个运算符进行计算.

}

else

{

while(s.Count!=0&& (OperatorPrecedence(inputStr[i]) <= OperatorPrecedence(s.Peek())))

{

n2 = m.Pop();

n1 = m.Pop();

c1 = s.Pop(); //若当前运算符的优先级不比前一个运算符高，按照从左到右进行运算原则，则应弹出上一个运算符先进行运算，再放入该运算符。

sum = Operat(n1, n2, c1); //根据不同运算符进行不同运算。

m.Push(sum); //将运算结果放入数字栈。

}

s.Push(inputStr[i]);

}

while(inputStr[i+1]=='-') //若运算符下一个字符为‘-’，则判断为负号，而不是减号。

{

t += "-";

i++;

}

}

}

while (s.Count != 0) //处理栈剩余运算符和运算数，直到符号栈为空，数字栈剩余最后一个数极为结果

{

n2 = m.Pop();

n1 = m.Pop();

c1 = s.Pop();

a = Operat(n1, n2, c1);

m.Push(a);

}

}

return a.ToString();

}

catch

{

string error = "Error"; //出现错误返回错误

return error;

}

}

public static int OperatorPrecedence(char a) //操作符优先级判断方法

{

int i = 0;

switch (a)

{

case '+': i = 1; break;

case '-': i = 1; break;

case '\*': i = 2; break;

case '/': i = 2; break;

case '^': i = 3; break;

case '!': i = 4; break;

}

return i;

}

public static double Operat(double n1, double n2, char c1) //加减乘除开方阶乘运算

{

double sum = 0;

switch (c1)

{

case '+': sum = n1 + n2; break;

case '-': sum = n1 - n2; break;

case '^': sum = Math.Pow(n1,n2); break;

case '\*': sum = n1 \* n2; break;

case '/': sum = n1 / n2; break;

case '!': sum = F(n1); break;

}

return sum;

}

public static double F(double n) //递归函数求阶乘

{

if (n != 0)

return n \* F(n - 1);

else

return 1;

}

}

}

* **案例2**

定义一个类ATM，实现银行ATM基本存款取款、以及获取账户各种信息的功能，密码连续输错三次将被自动锁号，使用继承得到一个子类，增加字段（如当前时间等）





编程：

Form1：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace 第七章\_案例2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

label4.Text = DateTime.Now.ToLongDateString(); //显示时间

label5.Text = DateTime.Now.ToLongTimeString();

timer1.Enabled = true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

label4.Text = DateTime.Now.ToLongDateString();

label5.Text = DateTime.Now.ToLongTimeString();

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e) //复选框【显示密码】p91

{

textBox2.PasswordChar = '\0';

}

private void button3\_Click(object sender, EventArgs e)

{

textBox1.Text = null;

textBox2.Text = null;

}

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

{

string[] a = new string[100];

string[] c = new string[7];

save.get(a, @"text2.txt");

int i=0;

if(textBox1.Text == null && textBox2.Text == null)

{

MessageBox.Show("请输入账号或密码");

}

if(textBox1.Text!=null&&textBox2.Text!=null)

{

if (textBox1.Text == a[0] && textBox2.Text == a[1])

{

操作界面 b = new 操作界面();

b.Show();

this.Hide();

for(i=0;i<7;i++)

{

c[i] = a[i];

}

save.put(c, @"message.txt");

}

else if (textBox1.Text == a[7] && textBox2.Text == a[8])

{

操作界面 b = new 操作界面();

b.Show();

this.Hide();

for (i = 0; i < 7; i++)

{

c[i] = a[i+7];

}

save.put(c, @"message.txt");

}

else

{

MessageBox.Show("输入账号或密码错误！");

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

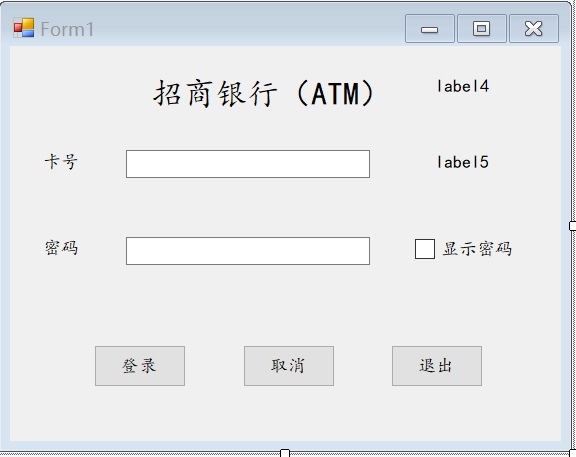
Application.Exit();

}

}

}

界面:



操作界面：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace 第七章\_案例2

{

public partial class 操作界面 : Form

{

public 操作界面()

{

InitializeComponent();

}

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

{

string[] a = new string[7];

save.get(a, @"message.txt");

textBox1.Text = a[2];

}

private void button6\_Click(object sender, EventArgs e)

{

Form1 a = new Form1();

a.Show();

this.Hide();

}

private void button7\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void timer1\_Tick(object sender, EventArgs e)

{

label6.Text = DateTime.Now.ToLongDateString();

label5.Text = DateTime.Now.ToLongTimeString();

}

private void 操作界面\_Load(object sender, EventArgs e)

{

label6.Text = DateTime.Now.ToLongDateString();

label5.Text = DateTime.Now.ToLongTimeString();

timer1.Enabled = true;

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = null;

}

private void button3\_Click(object sender, EventArgs e)

{

string[] a = new string[7];

string[] b = new string[14];

save.get(a, @"message.txt");

save.get(b, @"text2.txt");

double n = 0,m,i=0;

string q;

n = Convert.ToDouble(a[2]) - Convert.ToDouble(this.textBox2.Text);

q = Convert.ToString(n);

a[2] = q;

if (a[0]==b[0])

{

b[2] = a[2];

MessageBox.Show("取款成功");

save.put(a, @"message.txt");

save.put(b, @"text2.txt");

}

else if (a[0] == b[7])

{

b[9] = a[2];

MessageBox.Show("取款成功");

save.put(a, @"message.txt");

save.put(b, @"text2.txt");

}

}

private void button4\_Click(object sender, EventArgs e)

{

string[] a = new string[7];

save.get(a, @"message.txt");

listBox1.Items.Add(a[3]);

listBox1.Items.Add(a[4]);

listBox1.Items.Add(a[5]);

listBox1.Items.Add(a[6]);

}

private void button5\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

}

}

}

界面：



类：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace 第七章\_案例2

{

class save

{

public static void put(string [] old,string path)

{

int i = 0;

StreamWriter sw = new StreamWriter(path);

for(;i<old.Length ;i++ )

{

sw.WriteLine(old[i]);

}

sw.Close();

}

public static void get(string[] news,string path)

{

int i = 0;

string line = "";

StreamReader sr = new StreamReader(path);

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

{

news[i] = line;

i++;

}

sr.Close();

}

}

}

运行结果：



