**平 流 沉 淀 池 设 计 计 算**

班 级：水1102

学生姓名：别旭峰

学 号：U201115767

指导教师：罗凡

华 中 科 技 大 学（HUST）

2014年4月9日

目录

[1.计算方法 2](#_Toc384836504)

[2.总体设计与算法 5](#_Toc384836505)

[3.源程序 7](#_Toc384836506)

[4.程序计算结果 18](#_Toc384836507)

平流沉淀池（初沉池）设计计算

平流沉淀池是目前污水处理厂使用较多的一种池型，本设计采用面对对象设计中使用最为广泛的C#语言编写，能很好的实现可视化界面和人机交互。当用户输入主要参数后，软件能自动进行计算并给出结果，如有特殊要求需对次要参数进行调整，可人为输入次要参数，再由计算机给出计算结果，使用方便，能满足初高级用户的需求。此外，该软件对用户在使用过程中可能会出现的相关错误进行了优化，如禁止输入字母、其他无关数据以及输入的相关参数超出规范标准等问题，经过反复的优化，大量减少了软件出现错误的情况。

# 1.计算方法

参照现行手册及规范编制程序

设计要求：

1.池子的长宽比不小于4，以4~5为宜。当长宽比过小时，池内水流的均匀性差，容积效率低，影响沉降效果；大型沉淀池可考虑设导流墙。

2.池子的长深比不小于8，以8~12为宜。

3.沉淀池数目不少于两座，并应考虑一座发生故障时，另一座能负担全部流量的可能性。

表1：城市污水沉淀池设计数据及产生的污泥量表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 沉淀池类型 | 沉淀时间  （h） | 表面水力负荷  m3/(m2·h) | 污泥量 | | 污泥含水率  (%) |
| g/(p·d) | L/(p·d) |
| 初次沉淀池 | 1.0~2.0 | 1.5~3.0 | 14~27 | 0.36~0.83 | 95~97 |

计算公式

1.沉淀池表面积

q——表面水力负荷，即要求去除的颗粒沉速，如无试验资料，可参考表1选用。

2.沉淀部分有效水深

t——污水沉淀时间，初次沉淀池1~2h，参见表1。

3.沉淀部分有效容积

4.沉淀池长度

v——最大设计流量时的水平流速，mm/s，一般不大于5mm/s。

5.沉淀池宽度

6.单格沉淀池宽度

n1——沉淀池分格数

7.校核长宽比与长深比

长宽比L/b:不小于4，以4~5为宜

长深比L/h2:不小于8，以8~12为宜

8.污泥部分所需容积

V——每日污泥量，m3/d；

S——每人每日产生的污泥量，L/(p·d)。生活污水的污泥量见表1；

N——设计人口数；

T——两次排泥的时间间隔，初次沉淀池按2d考虑。

9.每格沉淀池污泥部分所需容积

10.污泥斗容积

污泥斗设在沉淀池的进水端，采用重力排泥，排泥管伸入污泥斗底部，为防止污泥斗底部积泥，污泥斗底部尺寸一般小于0.5m，本设计采用0.4m，污泥斗倾角为60o。

11.沉淀池高度

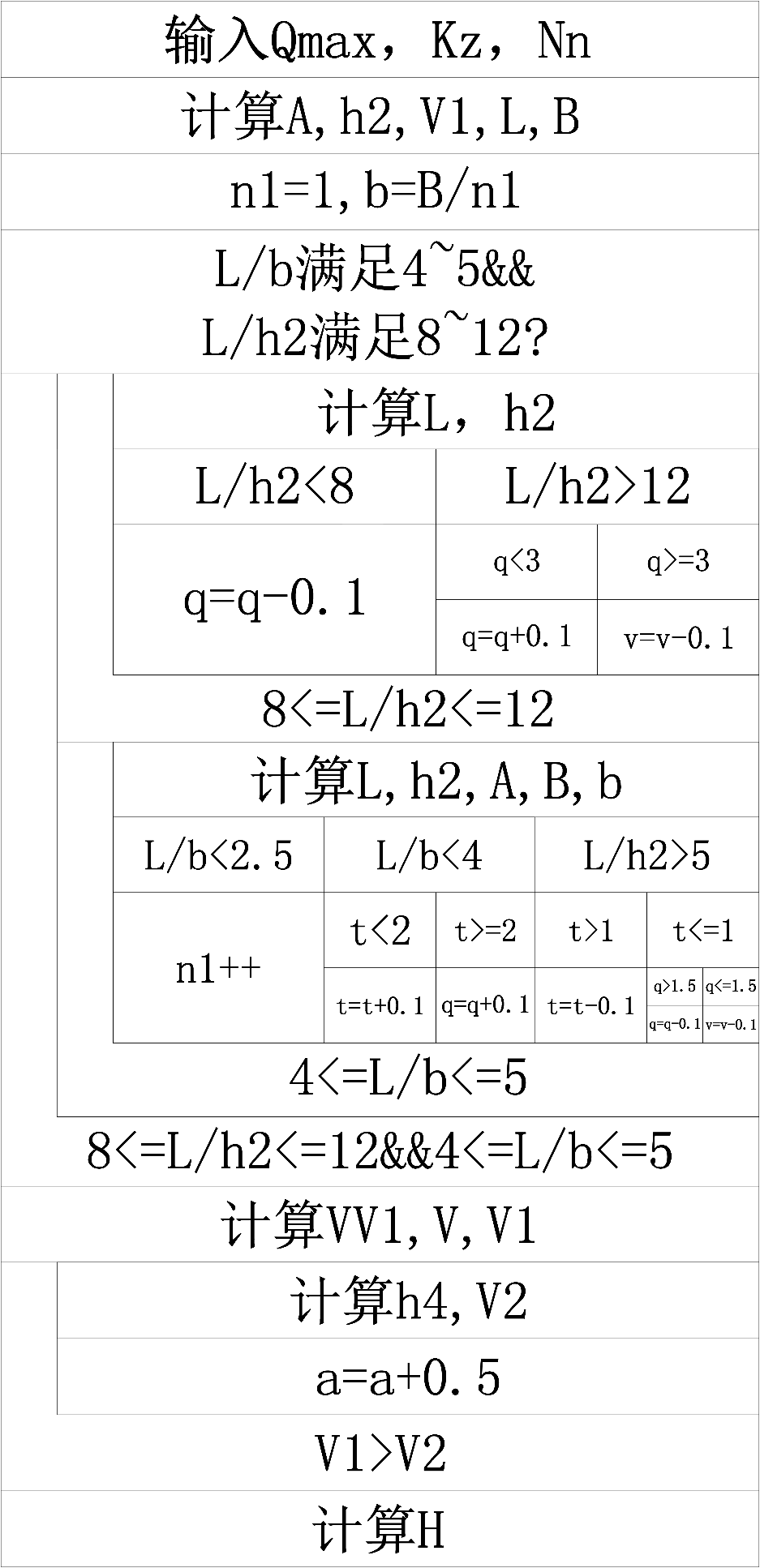
h1——超高，采用0.3m；

h2——沉淀区高度，m；

h3——缓冲区高度，当无刮泥机时，取0.5m；有刮泥机时，缓冲层的上缘应高出刮板0.3m；

h4——污泥区高度，m。

# 2.总体设计与算法



**相关参数：**

Qmax：设计流量

Nn：设计人口数

N：沉淀池个数

q：表面水力负荷

T：污水沉淀时间

v：最大设计流量时的水平流速

S：每人每日污泥量

h3：缓冲区高度

A：沉淀池面积

h2：沉淀区高度

VV1：沉淀部分有效容积

L：沉淀池长度

B：沉淀池宽度

b：单格沉淀池宽度

n1：沉淀池格数

V1：每格沉淀池污泥部分所需容积

V2：污泥斗容积

h4：污泥区高度

a1：污泥斗底部尺寸

a：污泥斗上部尺寸

H：总高度

**程序功能说明：**

本程序的亮点在于界面的设计和用户体验方面的完善，本软件具有极大的容错率。主要体现在：

1.在参数输入框内禁止输入字母、特殊符号以及两个及两个以上的点号。

2.当用户输入数据不符合规范时，会弹出提示对话框，告知范围。

3.用户输入数据后点回车键会自动进行计算。

4.计算结果区域禁止用户输入和更改数据。

此外，根据测试，该算法计算出的结果较为接近实际情况，计算结果合理可靠，且没有发现程序崩溃的现象。

# 3.源程序

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 WindowsFormsApplication3

{

public partial class pingliu : Form

{

public pingliu()

{

InitializeComponent();

}

double Qmax;//设计流量

double Q;

int N = 2;

double A;

double q = 2;//(1.5~3)

double t = 1.8;//(1~2)

double h2;//h2=q\*t

double VV1;//V1=Q\*t\*3600

double v = 5;//(<5)

double L;//L=v\*t\*3.6

double B;//B=A/L

double b;// L/b>4 (4~5)

int n1;//n1=B/b

// L/h2>8 (8~12)

double S = 0.6;//(0.36~0.83)

double Nn;//设计人口数

double T = 2;//T=2

double V;//V=S\*N\*T/1000n

double V1;//V1=V/n1

double V2;

double h3 = 0.5;//缓冲区高度

double h4;

double a1 = 0.4;//污泥斗底部尺寸

double a = 1;//污泥斗上部尺寸

double H;//总高度

int s1, s2;

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

{

}

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

{

Qmax = Convert.ToDouble(textBox1.Text);

if (Qmax <= 2300) { Qmax = 43200; textBox1.Text = Convert.ToString(Qmax); MessageBox.Show("设计流量大于2300！"); }

Nn = Convert.ToDouble(textBox3.Text);

if (Nn <= 0) { Nn = 31.3; textBox3.Text = Convert.ToString(Nn); MessageBox.Show("设计人口数大于0！"); }

N = Convert.ToInt32(textBox4.Text);

if (N < 2) { N = 2; textBox4.Text = Convert.ToString(N); MessageBox.Show("沉淀池个数大于等于2！"); }

q = Convert.ToDouble(textBox5.Text);

if (q > 3 || q < 1.5) { q = 2; textBox5.Text = string.Format("{0:f1}", q); MessageBox.Show("表面水力负荷的范围是1.5~3.0 m3/(m2\*h)！"); }

t = Convert.ToDouble(textBox6.Text);

if (t > 2 || t < 1) { t = 1.8; textBox6.Text = string.Format("{0:f1}", t); MessageBox.Show("污水沉淀时间的范围是1~2 h！"); }

v = Convert.ToDouble(textBox7.Text);

if (v > 5 || v <= 0) { v = 5; textBox7.Text = string.Format("{0:f1}", v); MessageBox.Show("最大设计流量的水平流速的范围是0~5 mm/s！"); }

S = Convert.ToDouble(textBox8.Text);

if (S > 0.83 || S < 0.36) { S = 0.6; textBox8.Text = string.Format("{0:f2}", S); MessageBox.Show("每人每日污泥量的范围是0.36~0.83 L/(人\*天)！"); }

h3 = Convert.ToDouble(textBox9.Text);

if (h3 <= 0) { h3 = 0.5; textBox9.Text = Convert.ToString(h3); MessageBox.Show("缓冲区高度大于0！"); }

Q = Qmax / N;

n1 = 1;

A = Q / 24 / q;

for (; ; )

{

for (; ; )

{

L = v \* t \* 3.6;

h2 = q \* t;

if (L / h2 >= 8 && L / h2 <= 12) { s1 = 0; break; }

else if (L / h2 < 8 && q > 1.5) q = q - 0.1;

else if (L / h2 < 8 && q < 1.5) { MessageBox.Show("error!"); s1 = 0; break; }

else if (L / h2 > 12 && q < 3) q = q + 0.1;

else if (L / h2 > 12 && q >= 3) v = v - 0.1;

else if (L / h2 > 12 && q > 3 && v < 3.4 ) { MessageBox.Show("error!"); s1 = 0; break; }

}

for (; ; )

{

L = v \* t \* 3.6;

h2 = q \* t;

A = Q / 24 / q;

B = A / L;

b = B / n1;

if (L / b >= 4 && L / b <= 5) { s2 = 0; break; }

else if (L / b < 2.5) n1++;

else if (L / b < 4 && t < 2) t = t + 0.1;

else if (L / b < 4 && t >= 2 && q <= 3) q = q + 0.1;

else if (L / b < 4 && t > 2 && q > 3) { MessageBox.Show("error!"); s2 = 0; break; }

else if (L / b > 5 && t > 1) t = t - 0.1;

else if (L / b > 5 && t <= 1 && q > 1.5) q = q - 0.1;

else if (L / b > 5 && t <= 1 && q <= 1.5 && v > 3.4) v = v - 0.1;

else if (L / b > 5 && t <= 1 && q <= 1.5 && v < 3.4) { MessageBox.Show("error!"); s2 = 0; break; }

}

if (s1 == 0 && s2 == 0) break;

}

VV1 = Q \* t / 24;

V = S \* Nn \* T \* 10 / N;

V1 = V / n1;

for (; ; )

{

h4 = (a - a1) \* Math.Sqrt(3) / 2;

V2 = h4 \* (a \* a + a1 \* a1 + a \* a1) / 3;

if (V2 > V1) break;

else a = a + 0.5;

}

H = 0.3 + h2 + h3 + h4;

textBox1.Text = string.Format("{0:f0}", Qmax);

textBox3.Text = string.Format("{0:f1}", Nn);

textBox4.Text = string.Format("{0:f0}", N);

textBox5.Text = string.Format("{0:f1}", q);

textBox6.Text = string.Format("{0:f1}", t);

textBox7.Text = string.Format("{0:f1}", v);

textBox8.Text = string.Format("{0:f1}", S);

textBox9.Text = string.Format("{0:f1}", h3);

richTextBox1.Text = "沉淀池表面积:" + string.Format("{0:f0}", A) + "m2。\n";

richTextBox1.Text += "沉淀部分有效容积:" + string.Format("{0:f0}", VV1) + "m3。\n";

richTextBox1.Text += "沉淀池长度:" + string.Format("{0:f1}", L) + "m。\n";

richTextBox1.Text += "沉淀池宽度:" + string.Format("{0:f1}", B) + "m。\n";

richTextBox1.Text += "沉淀池格数:" + n1 + "\n";

richTextBox1.Text += "单格沉淀池宽度:" + string.Format("{0:f1}", b) + "m。\n";

richTextBox1.Text += "校核长宽比:\n";

richTextBox1.Text += "长宽比:" + string.Format("{0:f1}", L / b) + " \n在4~5的范围内,符合要求。\n";

richTextBox1.Text += "长深比:" + string.Format("{0:f1}", L / h2) + " \n在8~12的范围内,符合要求。\n";

richTextBox1.Text += "污泥部分所需容积:" + string.Format("{0:f1}", V) + "m3。\n";

richTextBox1.Text += "单格污泥部分所需容积:" + string.Format("{0:f1}", V1) + "m3。\n";

richTextBox1.Text += "污泥斗容积:" + string.Format("{0:f1}", V2) + "m3。\n";

richTextBox1.Text += "清除污泥间隔时间:T=2d。\n";

richTextBox1.Text += "超高:" + 0.3 + "m。\n";

richTextBox1.Text += "沉淀部分有效水深:" + string.Format("{0:f1}", h2) + "m。\n";

richTextBox1.Text += "缓冲区高度:" + string.Format("{0:f1}", h3) + "m。\n";

richTextBox1.Text += "污泥区高度:" + string.Format("{0:f1}", h4) + "m。\n";

richTextBox1.Text += "沉淀池的总高度:" + string.Format("{0:f1}", H) + "m。\n";

}

private void textBox1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox2\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox3\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox4\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox5\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox6\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox7\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox8\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

private void textBox9\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)//如果输入的是回车键

{

this.button1\_Click(sender, e);//触发button事件

}

}

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

{

textBox1.Text = Convert.ToString(43200);

textBox3.Text = Convert.ToString(31.3);

textBox4.Text = Convert.ToString(2);

textBox5.Text = string.Format("{0:f1}", 2.0);

textBox6.Text = Convert.ToString(1.8);

textBox7.Text = string.Format("{0:f1}", 5.0);

textBox8.Text = Convert.ToString(0.6);

textBox9.Text = Convert.ToString(0.5);

richTextBox1.Clear();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox1.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox3.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox4.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox5\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox5.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox6\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox6.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox7\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox7.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox8\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox8.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

private void textBox9\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox9.Text;

e.Handled = e.KeyChar < '0' || e.KeyChar > '9'; //允许输入数字

if (e.KeyChar == (char)8) //允许输入回退键

{

e.Handled = false;

}

if (e.KeyChar == (char)46)

{

if (str == "") //第一个不允许输入小数点

{

e.Handled = true;

return;

}

else

{ //小数点不允许出现2次

foreach (char ch in str)

{

if (char.IsPunctuation(ch))

{

e.Handled = true;

return;

}

}

e.Handled = false;

}

}

}

}

}

# 4.程序计算结果

