****

**姓名：别旭峰**

**班级：给排水1102**

**学号：U201115767**

**时间：2014年4月**

**水工程程序设计**

目录

[1.化粪池计算选型 2](#_Toc385529546)

[1.1计算方法 2](#_Toc385529547)

[1.1.1设置条件 2](#_Toc385529548)

[1.1.2总容积计算 3](#_Toc385529549)

[1.1.3单个化粪池容积确定 4](#_Toc385529550)

[1.1.4尺寸规定 4](#_Toc385529551)

[1.1.5选型 5](#_Toc385529552)

[1.2总体设计与算法 5](#_Toc385529553)

[1.3源程序 6](#_Toc385529554)

[1.4程序计算结果 21](#_Toc385529555)

[2.平流沉淀池（初沉池）设计计算 22](#_Toc385529556)

[2.1计算方法 22](#_Toc385529557)

[2.2总体设计与算法 25](#_Toc385529558)

[2.3源程序 27](#_Toc385529559)

[2.4程序计算结果 39](#_Toc385529560)

# 1.化粪池计算选型

化粪池在建筑给水排水中使用广泛，且型号有固定规格，如设计人员每次设计时都查阅相关规范，不免费时费力，如为化粪池计算选型设计一个简易的小软件，设计人员只需输入相关数据即可得到结果，可以大大节省设计人员的时间和精力。现在网络上虽有几款可用于化粪池计算选型的软件，但经测试发现，它们大都是参照老版本的规范设计的，无法满足最新规范的要求。因此，我参照给水排水设计手册第二版第02册建筑给水排水使用C#语言编写了该软件。

## 1.1计算方法

### 1.1.1设置条件

在下列情况下应设置化粪池：

（1）当城镇没有污水处理厂时，生活粪便污水应设化粪池，经化粪池处理合格后的水方可排入城镇下水道或水体。

（2）城镇虽有生活污水处理厂的规划，但其建设滞后于建成生活小区，则应在生活小区内设置化粪池。

（3）一些大、中城市由于排水管网系统较长，为防止粪便淤积堵塞下水道，也要设化粪池。粪便污水井预处理后再排入城市管网。

（4）城市管网为合流制排水系统时，生活粪便污水应先经化粪池处理后，再排入合流制管网。

（5）大城市的排水管网对于排放水质有一定要求时，粪便污水也应设化粪池进行预处理。如化粪池处理后的水质仍不符合排放标准时，则需采用深化污水处理措施。

### 1.1.2总容积计算

化粪池总容积计算：

式中 V——化粪池总容积（m3）；

V1——污水部分容积（m3）；

V2——污泥部分容积（m3）；

V3——保护容积（m3）。

各部分容积计算如下：

（1）污水部分容积V1：

式中 N——化粪池实际使用人数，为总人数乘以系数α（%），α值与建筑物类型有关，见表1；

实际使用人数与总人数的百分比 表1

|  |  |
| --- | --- |
| 建筑物类型 | α值（%） |
| 医院、疗养院、幼儿园（有住宿） | 100 |
| 住宅、集体宿舍、旅馆 | 70 |
| 办公楼、教学楼、工业企业生活间 | 40 |
| 公共食堂、影剧院、体育场、其它类似公共场所（按座位计） | 10 |

q——每人每天的生活污水量L/(人·d)与用水量相同。如果粪便污水单独排出时，取20—30 L/(人·d)；当不同污水量定额的建筑物共用一个化粪池时，q值可按下式计算

其中 qn——各类建筑物污水量定额[L/(人·d)]；

Nn——相应建筑物污水量定额的实际使用人数；

t——污水在化粪池中的停留时间，根据污水量大小选用12~24h；当污水量较小或粪便污水单独排水时，选用上限值，反之可选用下限值。

（2）污泥部分容积V2：

式中 a——每人每天污泥量[L/(人·d)]，合流制排水时取0.7，粪便污水单独排放时取0.4；

T——污泥清掏周期(d)，根据污水温度和当地气候条件等因素，宜采用3个月~1a，当污水温度和当地气温较高时取下限值，反之取上限值；

b——进入化粪池中新鲜污泥的含水率，按95%计；

K——污泥发酵后体积缩减系数，按0.8计；

C——化粪池中发酵浓缩后污泥含水率，按90%计；

1.2——清掏污泥后按照遗留20%熟污泥量的容积系数。

（3）保护容积V3：根据化粪池容积大小，按照保护层高度为250~450mm设

计。

### 1.1.3单个化粪池容积确定

（1）当进入化粪池的污水量小于或等于10m3/d时，应选用双格化粪池。其中第一格容积应占总容积的75%。

（2）当进入化粪池的污水量大于10m3/d，应采用三格化粪池。第一格容积占总容积的50%，第二、三格容积各占25%。

（3）当化粪池的总容积超过50m3时，宜设置两个并联的化粪池。

（4）化粪池最小容积为2.0m3，此化粪池为圆形（又称化粪井）双格连通。每格有效直径不小于1.0m，两格容积相等。

### 1.1.4尺寸规定

化粪池的长度、深度、宽度应符合下列要求：

（1）由水面到池底的深度不得小于1.3m。

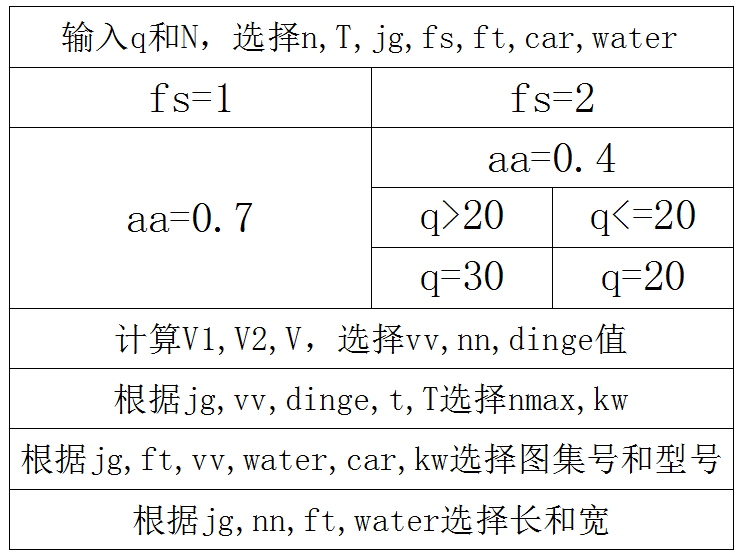
（2）池长不得小于1m。

（3）宽度不得小于0.75m。

### 1.1.5选型

给水排水标准图集（92S213、92S214）已经列出各种类型和规格的化粪池定型图，其规格从2~100m3。设计只须参照相关规范选用即可。

## 1.2总体设计与算法



相关参数：

n：建筑类型

q：污水定额

N：设计总人数

Nn：实际使用人数

t：水力停留时间

T：清掏周期

jg：结构

fs：排放方式

ft：是否有覆土选项

car：是否可过汽车选项

water：是否有地下水选项

V1：污水部分容积

V2：污泥部分容积

V：化粪池有效容积

nmax：最大允许使用人数

dinge：污水定额选择

rj：容积选项

nn：容积编号

kw：隔墙过水孔高度代号

## 1.3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 WindowsFormsApplication4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int i;

int n;//建筑类型

double aa;

double q;//污水定额

int N;//设计总人数

double Nn;//实际使用人数

int t;//水力停留时间:12小时，24小时，粪便污水单独排放选用24小时

int T;//清掏周期：90天，180天，360天

int jg;//结构：砖砌1，混凝土2

int fs;//排放方式：污废合流1，粪便单排2

int ft;//有覆土1，无覆土2

int car;//可过汽车1，不可过汽车2

int water;//有地下水1，无地下水2

double V, V1, V2;//化粪池容积

int dinge, rj1, nmax, kw;

int nn, rj;

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

{

n = comboBox1.SelectedIndex; //得到的i即为用户选择的选项的序号

N = Convert.ToInt32(textBox2.Text);

switch (n)

{

case 0: Nn = N \* 1; Nn = Convert.ToInt32(Nn); textBox3.Text = Convert.ToString(Nn); break;

case 1: Nn = N \* 0.7; Nn = Convert.ToInt32(Nn); textBox3.Text = Convert.ToString(Nn); break;

case 2: Nn = N \* 0.4; Nn = Convert.ToInt32(Nn); textBox3.Text = Convert.ToString(Nn); break;

case 3: Nn = N \* 0.1; Nn = Convert.ToInt32(Nn); textBox3.Text = Convert.ToString(Nn); break;

default: break;

}

q = Convert.ToInt32(textBox1.Text);

if (radioButton1.Checked == true)

{

t = 12;

}

if (radioButton2.Checked == true)

{

t = 24;

}

if (radioButton3.Checked == true)

{

T = 90;

}

else if (radioButton4.Checked == true)

{

T = 180;

}

else if (radioButton5.Checked == true)

{

T = 360;

}

if (radioButton6.Checked == true)

{

jg = 1;

}

if (radioButton7.Checked == true)

{

jg = 2;

}

if (radioButton8.Checked == true)

{

fs = 1; aa = 0.7;

}

if (radioButton9.Checked == true)

{

fs = 2; aa = 0.4;

if (q > 20) q = 30;

else q = 20;

}

if (checkBox1.Checked == true)

{

ft = 1;

}

else

{

ft = 2;

}

if (checkBox2.Checked == true)

{

car = 1;

}

else car = 2;

if (checkBox3.Checked == true)

{

water = 1;

}

else water = 2;

V1 = Nn \* q \* t / 24 / 1000;

V2 = aa \* Nn \* T \* (1.00 - 0.95) \* 0.8 \* 1.2 / (1 - 0.9) / 1000;

V = V1 + V2;

textBox4.Text = string.Format("{0:f3}", V);

int[] vv = { 2, 4, 6, 9, 12, 16, 20, 25, 30, 40, 50, 75, 100 };

for (i = 0; i < 13; i++)

{

if (V <= vv[i])

{

nn = i + 1;

rj = vv[i];

break;

}

}

//数据库

int[,] q500 = { { 7, 14, 21, 32, 43, 57, 71, 89, 107, 143, 178, 268, 357, 1 },

{ 6, 13, 19, 29, 39, 52, 64, 81, 97, 129, 161, 242, 322, 1 },

{ 5, 11, 16, 24, 32, 43, 54, 67, 81, 108, 135, 202, 270, 1 },

{ 4, 8, 11, 17, 23, 30, 38, 47, 57, 75, 94, 141, 189, 1 },

{ 4, 7, 11, 16, 21, 29, 36, 45, 54, 71, 89, 134, 178, 1 },

{ 3, 6, 10, 14, 19, 26, 32, 40, 48, 64, 81, 121, 161, 1 } };

int[,] q400 = { { 9, 17, 26, 39, 52, 69, 87, 109, 130, 174, 217, 326, 434, 1 },

{ 8, 15, 23, 35, 46, 61, 77, 96, 115, 154, 192, 288, 384, 1 },

{ 6, 12, 19, 28, 37, 50, 62, 78, 93, 125, 156, 234, 312, 1 },

{ 5, 9, 14, 21, 28, 37, 46, 58, 70, 93, 116, 174, 232, 1 },

{ 4, 9, 13, 20, 26, 35, 43, 54, 65, 87, 109, 163, 217, 1 },

{ 4, 8, 12, 17, 23, 31, 38, 48, 58, 77, 96, 144, 192, 1 }, };

int[,] q300 = { { 11, 22, 33, 50, 67, 89, 111, 139, 166, 222, 277, 416, 555, 1 },

{ 10, 19, 29, 43, 57, 76, 95, 119, 143, 190, 238, 356, 475, 1 },

{ 7, 15, 22, 33, 44, 50, 74, 92, 111, 148, 185, 277, 369, 1 },

{ 6, 12, 18, 27, 36, 48, 61, 76, 91, 121, 151, 227, 303, 1 },

{ 6, 11, 17, 25, 33, 44, 55, 69, 83, 111, 139, 208, 277, 1 },

{ 5, 10, 14, 21, 29, 38, 48, 59, 71, 95, 119, 178, 238, 1 } };

int[,] q250 = { { 13, 26, 39, 58, 77, 103, 129, 161, 193, 258, 322, 483, 644, 1 },

{ 11, 22, 32, 49, 65, 86, 108, 135, 162, 216, 270, 404, 539, 1 },

{ 8, 16, 24, 37, 49, 65, 81, 102, 122, 163, 203, 305, 407, 1 },

{ 7, 14, 21, 32, 43, 57, 71, 89, 107, 143, 178, 268, 357, 1 },

{ 6, 13, 19, 29, 39, 52, 64, 81, 97, 129, 161, 242, 322, 1 },

{ 5, 11, 16, 24, 32, 43, 54, 67, 81, 108, 135, 202, 270, 1 } };

int[,] q200 = { { 15, 31, 46, 69, 92, 123, 154, 192, 230, 307, 384, 576, 768, 1 },

{ 12, 25, 37, 56, 75, 100, 125, 156, 187, 249, 312, 467, 623, 1 },

{ 9, 18, 27, 41, 54, 72, 91, 113, 136, 181, 226, 339, 453, 1 },

{ 9, 17, 26, 39, 52, 69, 87, 109, 130, 174, 217, 326, 434, 1 },

{ 8, 15, 23, 35, 46, 61, 77, 96, 115, 154, 192, 288, 384, 1 },

{ 6, 12, 19, 28, 37, 50, 62, 78, 93, 125, 156, 234, 312, 1 }, };

int[,] q150 = { { 19, 38, 57, 86, 114, 152, 190, 238, 285, 380, 475, 713, 950, 1 },

{ 15, 30, 44, 66, 89, 118, 148, 185, 221, 295, 369, 554, 738, 1 },

{ 10, 20, 31, 46, 61, 82, 102, 128, 153, 204, 255, 383, 510, 2 },

{ 11, 22, 33, 50, 67, 89, 111, 139, 166, 222, 277, 416, 555, 1 },

{ 10, 19, 29, 43, 57, 76, 95, 119, 143, 190, 238, 356, 475, 1 },

{ 7, 15, 22, 33, 44, 59, 74, 92, 111, 148, 185, 277, 369, 1 }, };

int[,] q125 = { { 22, 43, 65, 97, 129, 173, 216, 270, 323, 431, 539, 809, 1078, 1 },

{ 16, 33, 49, 73, 98, 130, 163, 203, 244, 325, 407, 610, 813, 1 },

{ 11, 22, 33, 49, 65, 87, 109, 136, 164, 218, 273, 409, 545, 2 },

{ 12, 25, 37, 56, 75, 100, 125, 156, 187, 250, 312, 468, 624, 1 },

{ 11, 22, 32, 49, 65, 86, 108, 135, 162, 216, 270, 404, 539, 1 },

{ 8, 16, 24, 37, 49, 65, 81, 101, 122, 163, 203, 305, 407, 1 }, };

int[,] q100 = { { 25, 50, 75, 112, 150, 199, 249, 312, 374, 499, 623, 935, 1246, 1 },

{ 18, 36, 54, 81, 109, 145, 181, 226, 272, 362, 453, 679, 905, 1 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 178, 234, 292, 439, 585, 2 },

{ 15, 31, 46, 69, 92, 123, 154, 192, 230, 307, 384, 576, 768, 1 },

{ 12, 25, 37, 56, 75, 100, 125, 156, 187, 249, 312, 467, 623, 1 },

{ 9, 18, 27, 41, 54, 72, 91, 113, 136, 181, 226, 330, 453, 1 }, };

int[,] q50 = { { 36, 72, 109, 163, 217, 290, 362, 453, 543, 724, 905, 1358, 1810, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 351, 468, 585, 877, 1170, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 175, 234, 292, 439, 585, 2 },

{ 25, 50, 75, 112, 150, 199, 249, 312, 374, 499, 623, 935, 1246, 1 },

{ 18, 36, 54, 81, 109, 145, 181, 226, 272, 362, 453, 679, 905, 1 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 175, 234, 292, 439, 585, 2 }, };

int[,] q35 = { { 42, 84, 126, 189, 251, 335, 419, 524, 628, 838, 1047, 1571, 2095, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 385, 513, 641, 962, 1282, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 663, 2 },

{ 31, 61, 92, 138, 184, 245, 307, 383, 460, 613, 766, 1150, 1533, 1 },

{ 21, 42, 63, 94, 126, 168, 209, 262, 314, 419, 524, 786, 1047, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 192, 256, 321, 481, 641, 2 }, };

int[,] q25 = { { 46, 93, 139, 208, 278, 370, 463, 579, 702, 936, 1170, 1755, 2340, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 351, 468, 585, 877, 1170, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 661, 2 },

{ 40, 80, 119, 179, 239, 318, 398, 498, 579, 796, 995, 1493, 1990, 1 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 373, 497, 621, 932, 1243, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 661, 2 }, };

int[,] q20 = { { 46, 93, 139, 208, 278, 370, 463, 579, 746, 994, 1243, 1864, 2485, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 373, 497, 621, 932, 1243, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 661, 2 },

{ 40, 80, 119, 179, 239, 318, 398, 498, 597, 796, 995, 1493, 1990, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 397, 529, 661, 992, 1323, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 661, 2 }, };

int[,] q10 = { { 46, 93, 139, 208, 278, 370, 463, 579, 794, 1058, 1323, 1984, 2645, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 397, 529, 661, 992, 1323, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 661, 2 },

{ 46, 93, 139, 208, 278, 370, 463, 579, 746, 994, 1243, 1864, 2485, 2 },

{ 23, 46, 69, 104, 139, 185, 231, 289, 397, 529, 661, 992, 1323, 2 },

{ 12, 23, 35, 52, 69, 93, 116, 145, 198, 265, 331, 496, 661, 2 } };

int[,] qd30 = { { 62, 124, 186, 279, 372, 496, 620, 774, 929, 1239, 1549, 2323, 3098, 1 },

{ 40, 81, 121, 182, 242, 323, 404, 504, 605, 807, 1009, 1513, 2018, 2 },

{ 20, 41, 61, 91, 122, 162, 203, 253, 347, 463, 579, 868, 1157, 2 },

{ 42, 85, 127, 190, 254, 338, 423, 529, 635, 846, 1058, 1586, 2115, 1 },

{ 31, 62, 93, 139, 186, 248, 310, 387, 465, 620, 774, 1162, 1549, 1 },

{ 20, 40, 61, 91, 121, 161, 202, 252, 303, 404, 504, 757, 1009, 2 } };

int[,] qd20 = { { 73, 147, 220, 330, 440, 587, 733, 916, 1100, 1466, 1833, 2749, 3666, 2 },

{ 40, 81, 122, 182, 243, 324, 405, 506, 673, 898, 1122, 1683, 2244, 2 },

{ 20, 41, 61, 91, 122, 162, 203, 253, 347, 463, 579, 868, 1157, 2 },

{ 54, 107, 161, 241, 322, 429, 536, 671, 805, 1073, 1341, 2012, 2682, 1 },

{ 37, 73, 110, 165, 220, 293, 367, 458, 550, 733, 916, 1375, 1833, 2 },

{ 20, 41, 61, 91, 122, 162, 203, 253, 337, 449, 561, 842, 1122, 2 } };

int[] qq = { 500, 400, 300, 250, 200, 150, 125, 100, 50, 35, 25, 20, 10 };

//计算化粪池最大允许实际使用人数

if (fs == 1)

{

for (i = 0; i < 13; i++)

{

if (q >= qq[i])

{

dinge = qq[i];

break;

}

}

if (dinge == 500)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q500[rj1 - 1, nn - 1];

if (q500[rj1 - 1, 13] == 1) kw = 1;

else if (q500[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 400)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q400[rj1 - 1, nn - 1];

if (q400[rj1 - 1, 13] == 1) kw = 1;

else if (q400[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 300)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q300[rj1 - 1, nn - 1];

if (q300[rj1 - 1, 13] == 1) kw = 1;

else if (q300[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 250)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q250[rj1 - 1, nn - 1];

if (q250[rj1 - 1, 13] == 1) kw = 1;

else if (q250[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 200)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q200[rj1 - 1, nn - 1];

if (q200[rj1 - 1, 13] == 1) kw = 1;

else if (q200[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 150)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q150[rj1 - 1, nn - 1];

if (q150[rj1 - 1, 13] == 1) kw = 1;

else if (q150[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 125)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q125[rj1 - 1, nn - 1];

if (q125[rj1 - 1, 13] == 1) kw = 1;

else if (q125[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 100)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q100[rj1 - 1, nn - 1];

if (q100[rj1 - 1, 13] == 1) kw = 1;

else if (q100[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 50)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q50[rj1 - 1, nn - 1];

if (q50[rj1 - 1, 13] == 1) kw = 1;

else if (q50[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 35)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q35[rj1 - 1, nn - 1];

if (q35[rj1 - 1, 13] == 1) kw = 1;

else if (q35[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 25)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q25[rj1 - 1, nn - 1];

if (q25[rj1 - 1, 13] == 1) kw = 1;

else if (q25[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 20)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q20[rj1 - 1, nn - 1];

if (q20[rj1 - 1, 13] == 1) kw = 1;

else if (q20[rj1 - 1, 13] == 2) kw = 2;

}

else if (dinge == 10)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = q10[rj1 - 1, nn - 1];

if (q10[rj1 - 1, 13] == 1) kw = 1;

else if (q10[rj1 - 1, 13] == 2) kw = 2;

}

}

else if (fs == 2)

{

if (q >= 30)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = qd30[rj1 - 1, nn - 1];

if (qd30[rj1 - 1, 13] == 1) kw = 1;

else if (qd30[rj1 - 1, 13] == 2) kw = 2;

}

else if (q < 30)

{

if (t == 12 && T == 90) rj1 = 1;

else if (t == 12 && T == 180) rj1 = 2;

else if (t == 12 && T == 360) rj1 = 3;

else if (t == 24 && T == 90) rj1 = 4;

else if (t == 24 && T == 180) rj1 = 5;

else if (t == 24 && T == 360) rj1 = 6;

nmax = qd20[rj1 - 1, nn - 1];

if (qd20[rj1 - 1, 13] == 1) kw = 1;

else if (qd20[rj1 - 1, 13] == 2) kw = 2;

}

}

richTextBox1.Text = "选择化粪池容积编号为：" + nn + "\n";

richTextBox1.Text += "选择化粪池有效容积为:";

richTextBox1.Text += Convert.ToString(vv[nn - 1]);

richTextBox1.Text += "m3\n";

richTextBox1.Text += "化粪池最大允许实际使用人数:" + nmax + "人\n";

//计算完毕

//化粪池型号

richTextBox1.Text += "化粪池的型号为:" + nn + "-" + vv[nn - 1];

if (kw == 1) richTextBox1.Text += "A";

else if (kw == 2) richTextBox1.Text += "B";

if (water == 1) richTextBox1.Text += 1;

else if (water == 2) richTextBox1.Text += 0;

if (car == 1) richTextBox1.Text += 1;

else if (car == 2) richTextBox1.Text += 0;

richTextBox1.Text += "\n";

//标准图集号

if (jg == 1)

{

if (ft == 2)

{

if (nn >= 1 && nn <= 5) richTextBox1.Text += "标准图集:" + "92S213(一)" + "\n";

else if (nn >= 6 && nn <= 11) richTextBox1.Text += "标准图集:" + "92S213(二)" + "\n";

else if (nn >= 11) richTextBox1.Text += "标准图集:" + "92S213(五)" + "\n";

}

else if (ft == 1)

{

if (nn >= 1 && nn <= 5) richTextBox1.Text += "标准图集:" + "92S213(三)" + "\n";

else if (nn >= 6) richTextBox1.Text += "标准图集:" + "92S213(四)" + "\n";

}

}

else if (jg == 2)

{

if (ft == 2)

{

if (nn >= 1 && nn <= 5) richTextBox1.Text += "标准图集:" + "92S214(一)" + "\n";

else if (nn >= 6 && nn <= 11) richTextBox1.Text += "标准图集:" + "92S214(二)" + "\n";

else if (nn >= 11) richTextBox1.Text += "标准图集:" + "92S214(五)" + "\n";

}

else if (ft == 1)

{

if (nn >= 1 && nn <= 5) richTextBox1.Text += "标准图集:" + "92S214(三)\n";

else if (nn >= 6) richTextBox1.Text += "标准图集:" + "92S214(四)\n";

}

}

//占地外形尺寸

double[] lzqft0 = { 3.07, 5.28, 5.23, 6.47, 6.47, 7.92, 7.92, 7.92, 7.92, 9.32, 10.92, 9.76, 11.56 };

double[] kzqft0 = { 1.43, 1.69, 1.94, 2.44, 2.44, 2.94, 3.44, 3.44, 3.44, 3.44, 3.44, 6.31, 6.31 };

double[] lzqbft0 = { 3.07, 5.28, 5.23, 6.47, 6.47, 7.92, 7.92, 7.92, 7.92, 9.32, 10.92, 14.26, 15.46 };

double[] kzqbft0 = { 1.43, 1.69, 1.94, 2.44, 2.44, 2.94, 3.44, 3.44, 3.44, 3.44, 3.44, 3.68, 4.18 };

double[] lzqft1 = { 3.53, 5.48, 5.67, 6.81, 6.81, 8.26, 8.26, 8.26, 8.26, 9.66, 11.26, 10.20, 12.00 };

double[] kzqft1 = { 1.89, 1.89, 2.38, 2.88, 2.88, 3.38, 3.88, 3.88, 3.88, 3.88, 3.88, 6.75, 6.75 };

double[] lzqbft1 = { 3.53, 5.48, 5.67, 6.81, 6.81, 8.26, 8.26, 8.26, 8.26, 9.66, 11.26, 14.49, 15.69 };

double[] kzqbft1 = { 1.89, 1.89, 2.38, 2.88, 2.88, 3.38, 3.88, 3.88, 3.88, 3.88, 3.88, 4.14, 4.64 };

double[] lghft = { 2.90, 4.85, 4.80, 5.95, 5.95, 7.15, 7.15, 7.15, 7.15, 8.55, 10.15, 9.00, 10.80 };

double[] kghft = { 1.35, 1.35, 1.60, 2.10, 2.10, 2.60, 3.10, 3.10, 3.10, 3.10, 3.10, 5.80, 5.80 };

double[] lghbft = { 2.90, 4.85, 4.80, 5.95, 5.95, 7.15, 7.15, 7.15, 7.15, 8.55, 10.15, 13.35, 14.55 };

double[] kghbft = { 1.35, 1.35, 1.60, 2.10, 2.10, 2.60, 3.10, 3.10, 3.10, 3.10, 3.10, 3.20, 3.70 };

richTextBox1.Text += "化粪池占地外形尺寸：\n";

if (jg == 1)

{

if (ft == 1)

{

if (water == 2)

{

richTextBox1.Text += "长：" + lzqft0[nn - 1] + "m\n";

richTextBox1.Text += "宽：" + kzqft0[nn - 1] + "m\n";

}

else if (water == 1)

{

richTextBox1.Text += "长：" + lzqft1[nn - 1] + "m\n";

richTextBox1.Text += "宽：" + kzqft1[nn - 1] + "m\n";

}

}

else if (ft == 2)

{

if (water == 2)

{

richTextBox1.Text += "长：" + lzqbft0[nn - 1] + "m\n";

richTextBox1.Text += "宽：" + kzqbft0[nn - 1] + "m\n";

}

else if (water == 1)

{

richTextBox1.Text += "长：" + lzqbft1[nn - 1] + "m\n";

richTextBox1.Text += "宽：" + kzqbft1[nn - 1] + "m\n";

}

}

}

else if (jg == 2)

{

if (ft == 1)

{

richTextBox1.Text += "长：" + lghft[nn - 1] + "m\n";

richTextBox1.Text += "宽：" + kghft[nn - 1] + "m\n";

}

if (ft == 2)

{

richTextBox1.Text += "长：" + lghbft[nn - 1] + "m\n";

richTextBox1.Text += "宽：" + kghbft[nn - 1] + "m\n";

}

}

}

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

{

textBox1.Text = "200";

textBox2.Text = "150";

comboBox1.SelectedIndex = 1;

radioButton1.Checked = true;

radioButton3.Checked = true;

radioButton6.Checked = true;

radioButton8.Checked = true;

checkBox1.Checked = true;

checkBox2.Checked = true;

checkBox3.Checked = true;

textBox3.Text = "";

textBox4.Text = "";

richTextBox1.Text = "";

}

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

{

comboBox1.SelectedIndex = 1;

}

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 textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

string str = this.textBox2.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 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

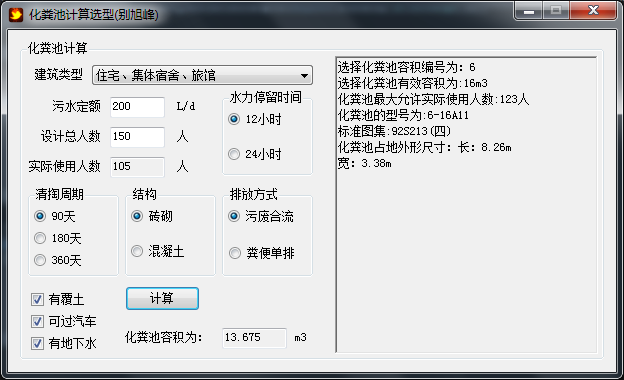
}

}

}

}

## 1.4程序计算结果



# 2.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。

## 2.1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

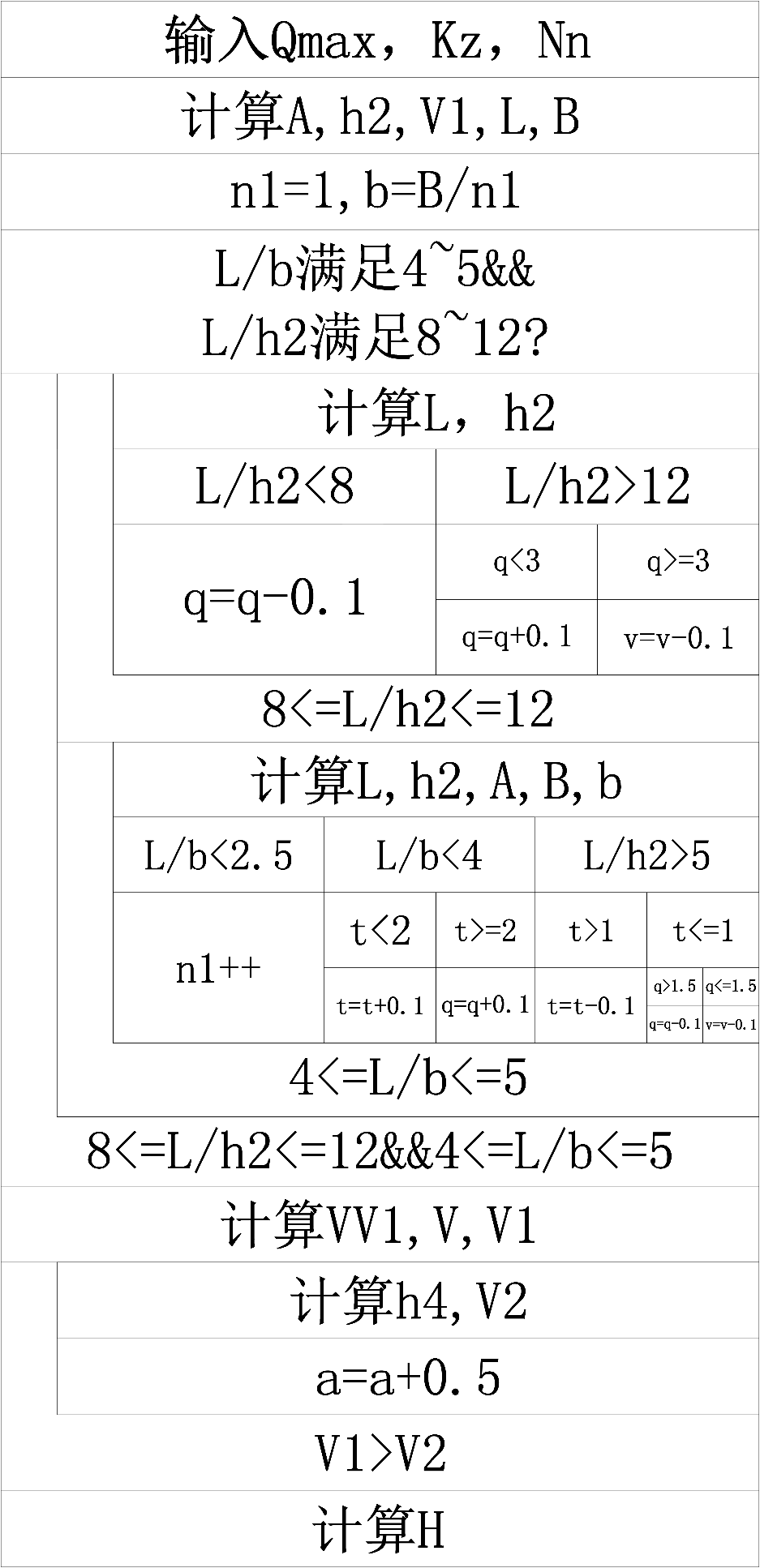
h1——超高，采用0.3m；

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。

## 2.2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。

## 2.3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;

}

}

}

}

}

## 2.4程序计算结果

