

Form1.cs

全局变量

//记录保留的符号，这些符号用来表示运算符或者数字。

```
enum A
```

```
{  
    e_Null,  
    e_Number,  
    e_Plus,  
    e_Minus,  
    e_Divide,  
    e_Multiply,  
    e_Sqrt,  
    e_Power,  
    e_LeftParen,  
    e_RightParen,  
    e_Sin,  
    e_Cos,  
    e_Tan,  
    e_Cot,  
    e_Log,  
    e_Lg,  
    e_Ln,  
    e_PI,  
    e_E  
};
```

//基本数学常量的定义

```
const double PI = 3.14159265358979323;  
const double MathE = 2.718281828459045;  
int curIndex = 0;  
double num;  
char charTemp;  
A sym = A.e_Null;
```

//错误信息提示

```
String[] errorMessage =  
{  
    "",  
    "你可能想要输入sin",//1  
    "不能识别的标识符",  
    "你可能想要输入sqrt",  
    "你可能想要输入cos",  
    "你可能想要输入cot",//5  
    "你可能想要输入tan",  
    "你可能想要输入log",  
    "^后需要跟()",  
    "括号对不匹配",  
    "sin,cos,sqrt,tan,cot后面需要跟()",//10  
    "log后面需要跟()",  
    "非法二进制",  
    "非法八进制",  
    "非法十进制",  
    "非法十六进制",//15  
    "你可能想要输log,ln或者lg",  
    "缺少括号",  
    "你可能想输入Pi,PI,pI或PI"
```

```

};
//被绑定的矩阵名称
string[] MatrixName = new string[50];
//被记录的矩阵
Matrix[] matrixs = new Matrix[1000];
//已经记录的矩阵的数量
int MatrixNum = 0;
//有关矩阵运算的保留的符号
enum B
{
    e_M_NULL,
    e_M_Plus,
    e_M_Minus,
    e_M_Mul,
    e_M_Inverse,
    e_M_LParen,
    e_M_RParen,
    e_M_Identifier
};

B symB = B.e_M_NULL;
string nameTemp = "";

```

由控件变化触发的事件

private void button[1~9]_Click(object sender, EventArgs e)

- 在输入框生成相应的数字

private void buttonDot_Click(object sender, EventArgs e)

- 在输入框生成点符号'.'

private void buttonEqual_Click(object sender, EventArgs e)

- 在输入框生成等号符号'='

private void buttonPlus_Click(object sender, EventArgs e)

- 在输入框生成加号符号'+'

private void buttonMinus_Click(object sender, EventArgs e)

- 在输入框生成减号符号'-'

private void buttonMultiply_Click(object sender, EventArgs e)

- 在输入框生成乘号符号'*'

private void buttonDivide_Click(object sender, EventArgs e)

- 在输入框生成除法符号'/'

private void buttonDelete_Click(object sender, EventArgs e)

- 删除输入框中光标前的一个字符。

private void buttonPower_Click(object sender, EventArgs e)

- 在输入框中生成幂运算符号'^()'

private void buttonSqrt_Click(object sender, EventArgs e)

- 在输入框中生成根号符号'sqrt()'

private void button[Sin,Cos,Tan,Cot]_Click(object sender, EventArgs e)

- 在输入框中生成三角函数符号'sin()', 'cos()', 'tan()', 'cot()'

private void buttonLog_Click(object sender, EventArgs e)

- 在输入框中生成对数运算符号'log()()'

private void buttonParen_Click(object sender, EventArgs e)

- 在输入框中生成一对括号'()'

private void buttonConvert_Click(object sender, EventArgs e)

- 当Convert按钮被按下触发事件。计算器会计算出进制转换的结果。

private void buttonPlot_Click(object sender, EventArgs e)

- 当Plot按钮被按下触发事件。计算器会绘制出对应函数的图像

private void textBox1_TextChanged(object sender, EventArgs e)

- 当textBox1中的内容改变时触发事件。

private void button12_Click(object sender, EventArgs e)

- 当“打开文件”按钮被按下时触发事件。计算器会打开一个对话框让客户选择需要打开的文件。

private void button13_Click(object sender, EventArgs e)

- 当Plot按钮被按下触发事件。计算器会绘制出对离散数据做插值后得到的函数的图像。

private void button14_Click(object sender, EventArgs e)

- 当Equal按钮被按下触发事件。计算器会计算出一个表达式的积分结果

private void buttonBound_Click(object sender, EventArgs e)

- 当Bound按钮被按下时触发事件。计算器会将用户提供的数据和它提供的字符串绑定在一起。

private void buttonClear_Click(object sender, EventArgs e)

- 当Clear按钮被按下时触发事件。初始化矩阵运算模块。

private void buttonMatrixEqual_Click(object sender, EventArgs e)

- 当'='按钮被按下时触发事件。计算出输入框中矩阵表达式的结果。

内部函数

double expr[i](string inputText)

- 递归计算出一个复杂的数学表达式的值

string Convert[i]To[j](string inputText)

- 将一个由进制i的数inputText转化为进制j表示的数。结果输出形式为字符串。

string DePreZero(string inputText)

- 删除一个表示数学值的字符串的前缀零。

public double AnalyseFunction(string inputText,string NumX)

- 给定一个含有未知变量的表达式inputText，该方法将数值NumX代入，计算出值

public double[] interpolation(double[] x, double[] y, int dataLength)

- 根据一组离散数据x[dataLength],y[dataLength],该方法计算出插值之后得到的标准多项式的各项系数。用数组存储。

public void Integral(double upperLimit, double floor, string func)

- 根据一个含有未知变量的表达式func,以及给定的上下限upperLimit,floor，调用相应方法绘制出它的图像

public Matrix MatrixExpression_[i](string inputText)

- 递归计算出一个复杂数学矩阵表达式的值

int position()

- 确定一个标识符是否已经记录（绑定了某个矩阵），如果存在返回一个非零位置，否则返回零。

Form2

由控件变化触发的事件

private void textBox[i]_TextChanged(object sender, EventArgs e)

- `textBox[i]`内容发生改变的时候触发事件。

内部函数

`public void chartAddSeries(Series s,double xMin, double xMax)`

- 根据点集s，横坐标上下限xMin,xMax绘制函数图象。