

GniMath 函数手册

张臻 冯昊明

2008年11月16日

目录

| | | |
|---|--------------------|----|
| 1 | MyError 类 | 2 |
| 2 | MyFunction 类 | 2 |
| 3 | T_Matrix 类 | 5 |
| 4 | StrComTree 类 | 7 |
| 5 | World2D 类 | 9 |
| 6 | World3D 类 | 13 |
| 7 | LinearProgramming类 | 15 |

第 1 节 MyError 类

该类为整个程序使用的自定义错误类，本类提供了两种错误构造方式，一种是按错误编号提供默认错误信息，另一种是自己初始化错误编号和错误信息。

对外部提供的函数接口：

```
public int GetErrNum()
```

功能：获取错误编码

```
public String GetErrMsg()
```

功能：获取错误信息

```
public int GetErrType()
```

功能：获取错误类型

第 2 节 MyFunction 类

该类用于放置全局使用的自定义函数。
提供的函数如下：

```
static public String MyTranStr(String str)
```

- 功能：转换字符串为标准字符串(半角)
- 参数：要转换的字符串
- 返回：转换完毕的字符串
- 错误：无

```
static public String CheckStr(String str)
```

- 功能：检查并返回非法字符
- 参数：要检查的字符串
- 返回：所含有的所有非法字符，返回空串则代表无非法字符
- 错误：无

```
static public String DelSpace(String str)
```

- 功能：删除字符串中的所有“空”字符（包括空格、回车、制表符）

- 参数: 要处理的字符串
- 返回: 处理完成的字符串
- 错误: 无

```
static public double IsNum(String NumStr) throws MyError
```

- 功能: 如果NumStr为一个数字则返回否则报错
- 参数: 要转化为数字的字符串
- 返回: 转化后的数字
- 错误: 该字符串不是数字

```
static public int Isint (String int Str) throws MyError
```

- 功能: 如果int Str为一个整型则返回否则报错
- 参数: 要转化为数字的字符串
- 返回: 转化后的数字
- 错误: 该字符串不是数字

```
static public String CutNumStr(double Num)
```

- 功能: 截取一个双精度浮点数(默认保留2位小数)
- 参数: 要截取的数字
- 返回: 截取后的字符串
- 错误: 无

```
static public String CutNumStr(double Num, int AP)
```

- 功能: 按要求保留精度截取一个双精度浮点数
- 参数: 要截取的数字和保留小数位数 (要求至少1位)
- 返回: 截取后的字符串
- 错误: 无

```
static public String CutNumStr(String TheStr)
```

- 功能: 截取一个数字字符串(默认保留2位小数)
- 参数: 要截取的数字字符串

- 返回: 截取后的字符串
- 错误: 无

```
static public String CutNumStr(String TheStr, int AP)
```

- 功能: 按要求保留精度截取一个数字字符串
- 参数: 要截取的数字字符串和保留小数位数 (要求至少1位)
- 返回: 截取后的字符串
- 错误: 无

```
static public double exp(double x)
```

- 功能: 求以e为底的指数
- 参数: 参数数字
- 返回: 计算结果
- 错误: 无

```
static public double log(double t)
```

- 功能: 求以e为底的对数
- 参数: 参数数字
- 返回: 计算结果
- 错误: 无

```
static public double pow(double x, double y)
```

- 功能: 求x的y次幂
- 参数: 两个参数数字
- 返回: 计算结果
- 错误: 无

```
static public double atan(double x)
```

- 功能: 反正切函数
- 参数: 参数数字
- 返回: 计算结果
- 错误: 无

```
static public double asin(double x)
```

- 功能: 反正弦函数
- 参数: 参数数字
- 返回: 计算结果
- 错误: 无

```
static public double acos(double x)
```

- 功能: 反余弦函数
- 参数: 参数数字
- 返回: 计算结果
- 错误: 无

第 3 节 T_Matrix 类

矩阵类除了类本身的矩阵元素及其行列外，还有矩阵的操作函数供外部调用。

内部变量:

```
public double [][] MatrixNum;//矩阵中各元素的值  
public int Rows; //矩阵的行数  
public int Cols; //矩阵的列数
```

构造函数:

```
public T_Matrix( String MaStr ) throws MyError
```

- 功能: 按信息构造一个矩阵
- 参数: 矩阵信息字符串
- 返回: --
- 错误: 矩阵信息不符合和格式

```
public T_Matrix(int MaRows, int MaCols)
```

- 功能: 构造一个指定行列的空矩阵 (元素均为0)
- 参数: 矩阵行数和列数
- 返回: --
- 错误: 无

```
public T_Matrix(double [] [] Num, int MaRows, int MaCols)
```

- 功能: 按数据构造一个指定行列的矩阵 (元素与Num相同)
- 参数: 矩阵行数和列数
- 返回: --
- 错误: 无

供外部的接口函数:

```
static public T_Matrix MatrixAdd(T_Matrix M1, T_Matrix M2) throws  
MyError
```

- 功能: 计算矩阵M1与M2的和
- 参数: 矩阵M1和M2
- 返回: 计算结果矩阵
- 错误: 两个矩阵的行或列不相同

```
static public T_Matrix MatrixSub(T_Matrix M1, T_Matrix M2) throws  
MyError
```

- 功能: 计算矩阵M1与M2的差
- 参数: 矩阵M1和M2
- 返回: 计算结果矩阵
- 错误: 两个矩阵的行或列不相同

```
static public T_Matrix MatrixMul(T_Matrix M1, T_Matrix M2) throws  
MyError
```

- 功能: 计算矩阵M1与M2的乘积
- 参数: 矩阵M1和M2
- 返回: 计算结果矩阵
- 错误: M1的列不等于M2的行

```
static public T_Matrix MatrixFactorial(T_Matrix M, int Num) throws  
MyError
```

- 功能: 计算矩阵M的乘幂
- 参数: 矩阵M及幂数
- 返回: 计算结果矩阵

- 错误: M非方阵, 阶数过大, 阶数小于1

```
static public T_Matrix MatrixT(T_Matrix M)
```

- 功能: 计算矩阵M的转置
- 参数: 矩阵M
- 返回: 计算结果矩阵
- 错误: 无

```
static public T_Matrix MatrixMulConst(T_Matrix M, double constNum)
```

- 功能: 计算矩阵M乘以常量
- 参数: 矩阵M及乘数
- 返回: 计算结果矩阵
- 错误: 无

```
static public T_Matrix MatrixR(T_Matrix M)
```

- 功能: 计算方阵M的逆矩阵
- 参数: 方阵M (需要在外部判断行列是否相等)
- 返回: 计算结果矩阵, 不存在则返回null
- 错误: 无

```
static public double MatrixDet(T_Matrix M)
```

- 功能: 计算方阵M对应行列式的值
- 参数: 方阵M (需要在外部判断行列是否相等)
- 返回: 计算结果, 不存在则返回null
- 错误: 无

第 4 节 StrComTree 类

该类用于把一个表达式字符串转化为表达式树, 并可以计算相应的表达式值。该表达式生成后可以对其注册变量和常量, 当计算表达式的是后将会替换对应的字符为注册的值。

内部变量:

```
public String ComStr; //保存表达式字符串

public T_ListNode Head; //保存生成单词的链表（链表头）

public T_TreeNode Root; //保存生成的表达式计算树（树根）

public T_VarList Vars; //变量链表

public T_MyStack NumStack; //数字堆栈

public T_MyStack SymStack; //符号堆栈

public String [] FuncStr; //符号堆栈

public String [] SymStack; //符号堆栈
```

构造函数:

```
public StrComTree(String ComputeStr) throws MyError
```

参数ComputeStr是需要转换的字符串，构造函数将执行四个步骤:

- 1. 检查字符串，如果有不可识别的字符扔出错误。调用函数MyFunction.CheckStr。
- 2. 多字符串进行检查扫描，转化为单词链。每个单词链节点包含，节点字符串和单词类型。调用函数CreateList, CheckList, SetList。
- 3. 把单词链转换为表达式树，通过堆栈分析单词链完成树转化，并进行简单的树优化(常数合并)。调用函数CreateTree, OptimizeTree。
- 4. 设变量表为空。执行语句Vars=null。

供外部的接口函数:

```
public double ComputeTree(T_TreeNode NowRoot) throws MyError
```

- 功能: 计算表达式树
- 参数: 要计算的树根
- 返回: 计算结果
- 错误: 表达式树构造失败（多种可能的错误）

```
public void AddVar(String Str) throws MyError
```

- 功能: 添加变量（仅字符，用于注册变量）
- 参数: 注册变量字符

- 返回: 无
- 错误: 添加重复变量

```
public void AddVar(String Str,double Num)throws MyError
```

- 功能: 添加变量 (仅字符, 用于注册变量)
- 参数: 注册变量字符及其值
- 返回: 无
- 错误: 添加重复变量

```
public void SetVar(String Str,double Num)throws MyError
```

- 功能: 设定变量值
- 参数: 要设定的变量字符及其值
- 返回: 无
- 错误: 不存在该变量

```
public void DelVar(String Str)throws MyError
```

- 功能: 删除变量
- 参数: 要删除的变量字符
- 返回: 无
- 错误: 不存在该变量

```
public boolean HaveVar(String Str)
```

- 功能: 判断是否有变量
- 参数: 检查的变量字符
- 返回: 是否有该变量
- 错误: 无

第 5 节 World2D 类

该类提供二维坐标系, 主要用作父类
内部变量:

```
protected double Xup; //绘制的x的范围上限
```

```
protected double Xdown; //绘制的x的范围下限
protected double Ymax; //绘制的y的范围上限
protected double Ymin; //绘制的y的范围下限
private double dfx; //fx点值误差
private double dfy; //fy点值误差
private int Window_W; //屏幕的宽度
private int Window_H; //屏幕的高度
private int Axis_X; //y轴屏幕x值
private int Axis_Y; //x轴屏幕y值
protected boolean ChangeMode=true; //坐标变换模式
```

构造函数:

```
public World2D()
```

- 功能: 初始化一个2D坐标系, 获取屏幕的宽和高
- 参数: 无
- 返回: --
- 错误: 无

供子类的接口函数:

```
protected double SxtoFx(double Sx)
```

- 功能: 由屏幕x坐标获取函数x值
- 参数: 屏幕x坐标
- 返回: 函数x值
- 错误: 无

```
protected double SytoFy(double Sy)
```

- 功能: 由屏幕y坐标获取函数y值
- 参数: 屏幕y坐标
- 返回: 函数y值

- 错误: 无

`protected double FxtoSx(double Fx)`

- 功能: 由函数x值获取屏幕x坐标
- 参数: 函数x值
- 返回: 屏幕x坐标
- 错误: 无

`protected double FytoSy(double Fy)`

- 功能: 由函数y值获取屏幕y坐标
- 参数: 函数y值
- 返回: 屏幕y坐标
- 错误: 无

`protected void InitScreen(Graphics g)`

- 功能: 初始化屏幕为背景色
- 参数: `Graphics`
- 返回: 无
- 错误: 无

`protected void DrawLineXY(Graphics g)`

- 功能: 绘制坐标轴
- 参数: `Graphics`
- 返回: 无
- 错误: 无

`protected void SetColor(Graphics g,int Type)`

- 功能: 设定当前的颜色
- 参数: `Graphics` 及要设定的颜色类型
- 返回: 无
- 错误: 无

```
protected void SetRangeTitle()
```

- 功能: 设定标题栏为显示范围
- 参数: Graphics 及要设定的颜色类型
- 返回: 无
- 错误: 无

```
protected void SetPressTitle(int Sx,int Sy)
```

- 功能: 设定标题栏为点选的坐标值
- 参数: 点选的屏幕x和y
- 返回: 无
- 错误: 无

```
protected void ChangeRange(int Key)
```

- 功能: 根据按下按键改变显示范围
- 参数: 按下的按键
- 返回: 无
- 错误: 无

虚函数重构功能:

```
protected void paint (Graphics g)
```

设定标题, 初始化背景并绘制坐标轴

```
public void pointerDragged(int x,int y)
```

显示选中点的坐标

```
public void pointerPressed(int x,int y)
```

显示选中点的坐标”

```
public void pointerReleased(int x,int y)
```

恢复标题的范围显示

```
public void keyReleased(int key)
```

调用ChangeRange改变显示参数

第 6 节 World3D 类

内部变量：

```
protected boolean ChangeMode=true; //模式选择
protected boolean Pressed=false; //触摸按下
protected double Proportion; //缩放比例
protected int Xplus,Yplus; //坐标平移量
protected double Angle1,Angle2; //角度参数
protected int OldPressX,OldPressY; //最早按下的屏幕坐标
protected double AngleBox1,AngleBox2; //临时角度参数
```

构造函数：

```
public World3D()
```

- 功能：初始化一个3D坐标系，获取屏幕的宽和高，初始化显示参数
- 参数：无
- 返回：--
- 错误：无

供子类的接口函数：

```
protected double GetSx(double fx,double fy,double fz)
```

- 功能：将三位坐标转化为二维，获取二维的x坐标
- 参数：三位坐标x, y, z
- 返回：二维坐标下的x
- 错误：无

```
protected double GetSy(double fx,double fy,double fz)
```

- 功能：将三位坐标转化为二维，获取二维的y坐标
- 参数：三位坐标x, y, z

- 返回: 二维坐标下的y
- 错误: 无

```
protected void Line3D(Graphics g,double x1,double y1,double  
z1,double x2,double y2,double z2)
```

- 功能: 三维直线绘制
- 参数: `Graphics` , 起点三维坐标和终点三维坐标
- 返回: 屏幕x坐标
- 错误: 无

```
protected void InitScreen(Graphics g)
```

- 功能: 初始化屏幕为背景色
- 参数: `Graphics`
- 返回: 无
- 错误: 无

```
protected void SetColor(Graphics g,int Type)
```

- 功能: 设定当前的颜色
- 参数: `Graphics` 及要设定的颜色类型
- 返回: 无
- 错误: 无

```
protected void ChangeRange(int Key)
```

- 功能: 根据按下按键改变显示参数
- 参数: 按下的按键
- 返回: 无
- 错误: 无

```
protected void DrawBox(Graphics g)
```

- 功能: 绘制单位测试盒 (用于拖动旋转)
- 参数: `Graphics`

- 返回: 无
- 错误: 无

虚函数重构功能:

`protected void paint (Graphics g)` 绘制一个黑色单位3D盒, 点击时绘制红色的单位3D盒

`public void pointerDragged(int x,int y)` 拖动旋转重新设定视角

`public void pointerPressed(int x,int y)` 记录旋转原始位置 (供计算新的视角)

`public void pointerReleased(int x,int y)` 用新视角替换老视角

`public void keyReleased(int key)` 调用ChangeRange改变显示参数

第 7 节

LinearProgramming类

公共变量:

`public double x[];` //线性规划取得最值时各变量的取值

`public double v;` //线性规划的最值

构造函数: `LinearProgramming(double A[][], double b[], double c[])`
`throws MyError`

功能: 计算一个线性规划的最大值

参数: 线性规划的不等式表示

返回: 无

错误: 线性规划不可满足, 想行规划最优解为 ∞

索引

acos, 5
AddVar, 8, 9
Angle1, 13
Angle2, 13
AngleBox1, 13
AngleBox2, 13
asin, 4
atan, 4
Axis_X, 10
Axis_Y, 10

ChangeMode, 10, 13
ChangeRange, 12, 14
CheckStr, 2
Cols, 5
ComputeTree, 8
ComStr, 8
CutNumStr, 3, 4

DelSpace, 2
DelVar, 9
dfx, 10
dfy, 10
DrawBox, 14
DrawLineXY, 11

exp, 4

FuncStr, 8
FxtoSx, 11
FytoSy, 11

GetErrMsg, 2
GetErrNum, 2
GetErrType, 2
GetSx, 13
GetSy, 13

HaveVar, 9
Head, 8

InitScreen, 11, 14
Isint, 3
IsNum, 3

keyReleased, 12, 15

Line3D, 14
LinearProgramming, 15
log, 4

MatrixAdd, 6
MatrixDet, 7
MatrixFactorial, 6
MatrixMul, 6
MatrixMulConst, 7
MatrixNum, 5
MatrixR, 7
MatrixSub, 6
MatrixT, 7
MyError, 2
MyFunction, 2
MyTranStr, 2

NumStack, 8

OldPressX, 13
OldPressY, 13

paint, 12, 15
pointerDragged, 12, 15
pointerPressed, 12, 15
pointerReleased, 12, 15
pow, 4
Pressed, 13
Proportion, 13

Root, 8
Rows, 5

SetColor, 11, 14
SetPressTitle, 12

SetRangeTitle, 12
SetVar, 9
StrComTree, 7, 8
SxtoFx, 10
SymStack, 8
SymStrArr, 8
SytoFy, 10

T_Matrix, 5, 6

v, 15
Vars, 8

Window_H, 10
Window_W, 10
World2D, 10
World3D, 13

x, 15
Xdown, 10
Xplus, 13
Xup, 9

Ymax, 10
Ymin, 10
Yplus, 13