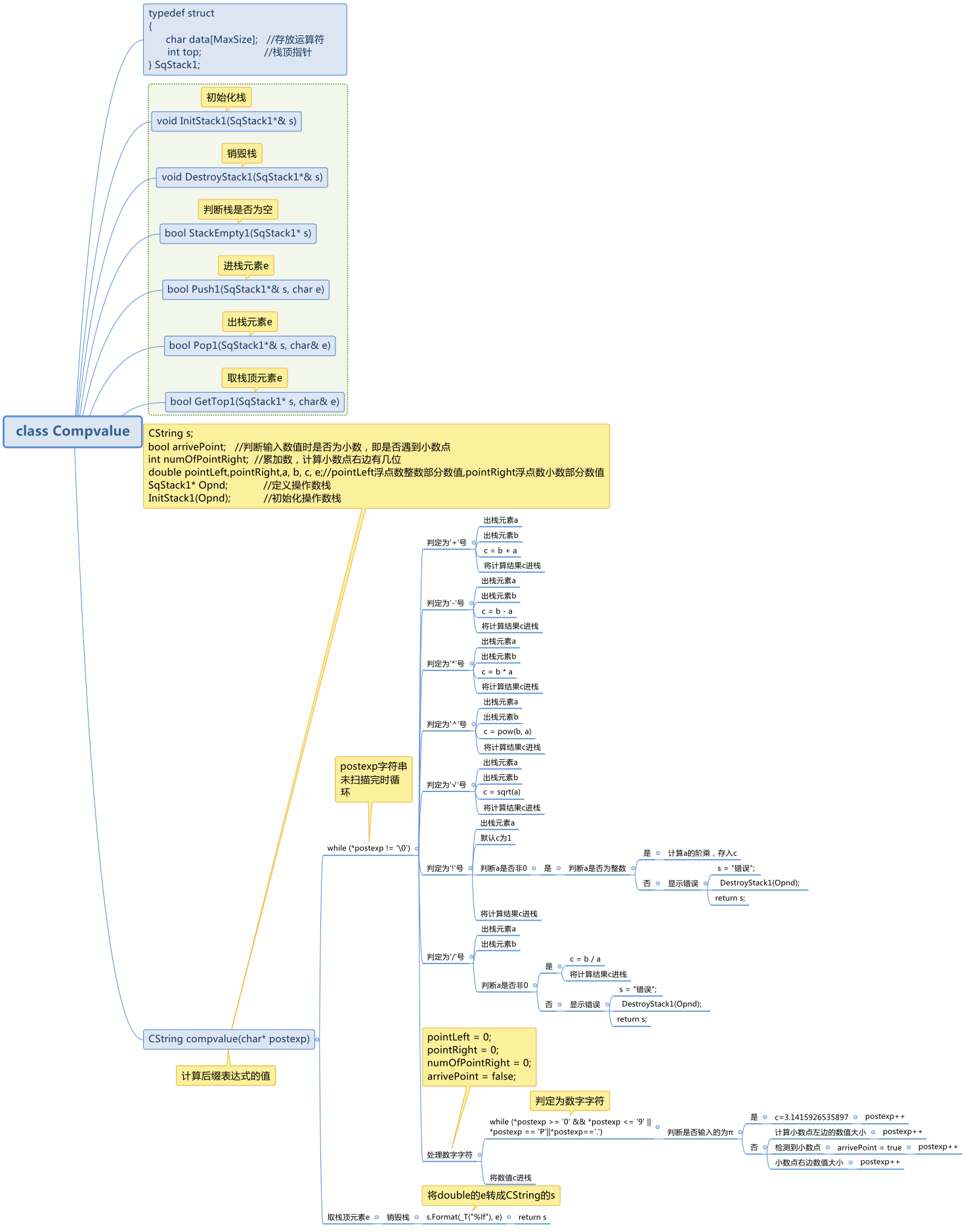
typedef struct
{
char data[MaxSize]; //存放运算符
int top; //栈顶指针
} SqStack1;
初始化栈
void InitStack1(SqStack1*& s)
销毁栈
void DestroyStack1(SqStack1*& s)
判断栈是否为空
bool StackEmpty1(SqStack1* s)
进栈元素e
bool Push1(SqStack1*& s, char e)
出栈元素e
bool Pop1(SqStack1*& s, char& e)
取栈顶元素e
bool GetTop1(SqStack1* s, char& e)
class Compvalue
CString s;
bool arrivePoint; //判断输入数值时是否为小数，即是否遇到小数点
int numOfPointRight; //累加数，计算小数点右边有几位
double pointLeft,pointRight,a, b, c, e;//pointLeft浮点数整数部分数值,pointRight浮点数小数部分数值
SqStack1* Opnd; //定义操作数栈
InitStack1(Opnd); //初始化操作数栈
CString compvalue(char* postexp)
计算后缀表达式的值
while (*postexp != '\0')
postexp字符串未扫描完时循环
判定为'+'号
出栈元素a
出栈元素b
c = b + a
将计算结果c进栈
判定为'-'号
出栈元素a
出栈元素b
c = b - a
将计算结果c进栈
判定为'*'号
出栈元素a
出栈元素b
c = b * a
将计算结果c进栈
判定为'^'号
出栈元素a
出栈元素b
c = pow(b, a)
将计算结果c进栈
判定为'√'号
出栈元素a
出栈元素b
c = sqrt(a)
将计算结果c进栈
判定为'!'号
出栈元素a
默认c为1
判断a是否非0
是
判断a是否为整数
是
计算a的阶乘，存入c
否
显示错误
s = "错误";
DestroyStack1(Opnd);
return s;
将计算结果c进栈
判定为'/'号
出栈元素a
出栈元素b
判断a是否非0
是
c = b / a
将计算结果c进栈
否
显示错误
s = "错误";
DestroyStack1(Opnd);
return s;
pointLeft = 0;
pointRight = 0;
numOfPointRight = 0;
arrivePoint = false;
判定为数字字符
处理数字字符
while (*postexp >= '0' && *postexp <= '9' || *postexp == 'P'||*postexp=='.')
判断是否输入的为π
是
c=3.1415926535897
postexp++
否
计算小数点左边的数值大小
postexp++
检测到小数点
arrivePoint = true
postexp++
小数点右边数值大小
postexp++
将数值c进栈
将double的e转成CString的s
取栈顶元素e
销毁栈
s.Format(_T("%lf"), e)
return s