

## TB 编程入门

第一章 TB 公式基础知识.....	2
1.1 什么是 TB 公式? .....	2
1.2 TB 的运行机制.....	3
1.3 TB 公式结构.....	3
第二章 TB 公式的具体实现.....	3
2.1 TB 公式从编写到运行.....	4
2.2 TB 公式的基本知识点.....	6
2.2.1 TB 公式中有三种基本的数据类型.....	7
2.2.2 序列变量与普通变量的比较.....	7
2.2.3 参数的声明和使用.....	7
2.2.4 变量和变量的声明.....	8
2.2.5 变量的赋值（变量类型和表达式的类型要一致） .....	8
2.3.1 参数的作用.....	10
2.3.2 控制语句.....	12
第三章 交易策略参数优化.....	17
3.1 参数优化基本操作.....	17
3.2 优化结果生成分析图表.....	18

## 第一章 TB 公式基础知识

### 1.1 什么是 TB 公式？

TB 公式类型：①公式运用（包括技术指标，交易指令等） ②用户函数

注意：这两者的主要结构是一样的，一些细微的差别我们在后续章节中会点到。

对于初学者而言，先不用深究这两者的区别。

通过调用 TB 公式，您可以在交易开拓者中进行技术分析、交易策略优化测试、公式报警、自动交易等操作。TB 公式的计算，都是建立在基本数据源 (Bar 数据，即 k 线) 之上。

#### Bar 数据

Bar 数据是 TB 公式运行的基础。当前时间周期下所有 K 线的相关数据，按照时间从先到后的顺序排列而成的序列数据。每根 K 线中包含的数据如下：

Bar数据	含义
Date	当前K线的日期
Time	当前K线的开始时间
Open	当前K线的开盘价
High	当前K线的最高价
Low	当前K线的最低价
Close	当前K线的收盘价（最新价）
Vol	当前K线成交量
OpenInt	当前K线持仓量
CurrentBar	当前K线的索引值（K线的编号，从0开始）
BarStatus	当前K线的状态值（0—第一根K线、2—最后即最新一根K线、1—其他K线）

举例：

比较今天的最高价是否突破了昨天的最高价

表达式为：High > High[1]

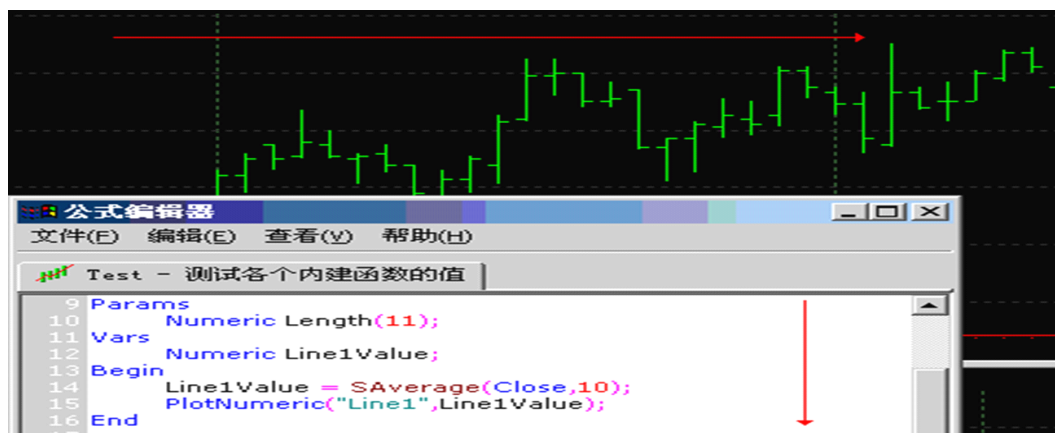
比较今天的最低价是否突破了前两天的最低价

表达式为：low > low[1] and low>low[2] （and 也可以用&&表示）

bar 数据就是序列数据，可以进行回溯读取。什么是序列数据，我们将在后续的章节中介绍。这里，我们只需记住 bar 数据是序列数据就可以了。

## 1.2 TB 的运行机制

从左到右，从上到下：



## 1.3 TB 公式结构

TB 的公式一般由三段组成。

Params

Numeric Length(10);

... ..

Vars

NumericSeries MA;

... ..

Begin

MA = AverageFC(Close, Length);

... ..

End

公式参数段

公式变量段

公式脚本段

## 第二章 TB 公式的具体实现

Tips:这一章我们将结合两个例子为大家呈现 TB 公式的具体实现。

例子一旨在让大家熟悉 TB 公式从编写到运行的操作流程，初学者可以对应界面截图跟着做。

例子二则是 TB 公式的代码的具体编写流程，旨在让大家熟悉公式编写的几个步骤。

例子三是一个简单的交易策略实现。接着我们会就编码的知识点有一个系统的概括，供大家理解和查阅，也希望大家通过本例对 TB 公式的结构有一个清晰的印象。

以下模型源码内容仅供编写参考使用，如用于交易使用，风险自负。

## 2.1 TB 公式从编写到运行

例 1: Hello World

目标: 在文件指定文件(本例中是 sample1.txt 这个文件)中写一行字符串:hello world

程序如下:

Begin

```
FileAppend("c:\\tb\\sample1.txt", "Hello World!");
```

End

大家可在 TB 系统中查阅到实现该目标功能的函数，如下图:

### 附录B - 系统函数 - FileAppend

#### FileAppend

说明	在指定文件中追加一行字符串。
语法	Bool FileAppend(String strPath,String strText)
参数	strPath 指定文件的路径，请使用全路径表示，并使用\\做路径分割符，否则会执行失败 strText 输出的字符串内容
备注	在指定文件中追加一行字符串，返回值为布尔型。 执行成功返回True，执行失败返回False。
示例	FileAppend("C:\\Formula.log","Close = "+Text(Close));

实验步骤:

TB 公式→新建公式应用 →输入公式简称 →选择适当的模板;

Step 1: 新建公式应用

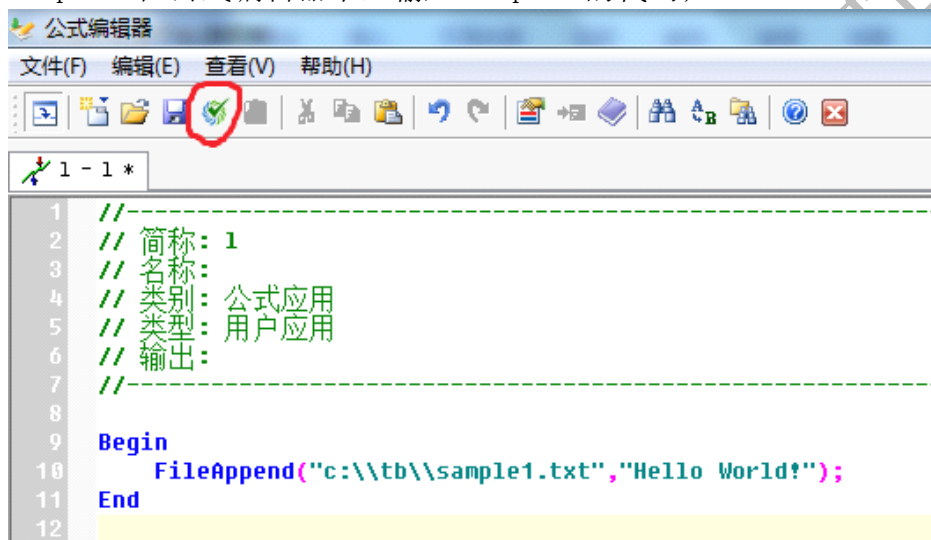




Step 2: 输入公式简称

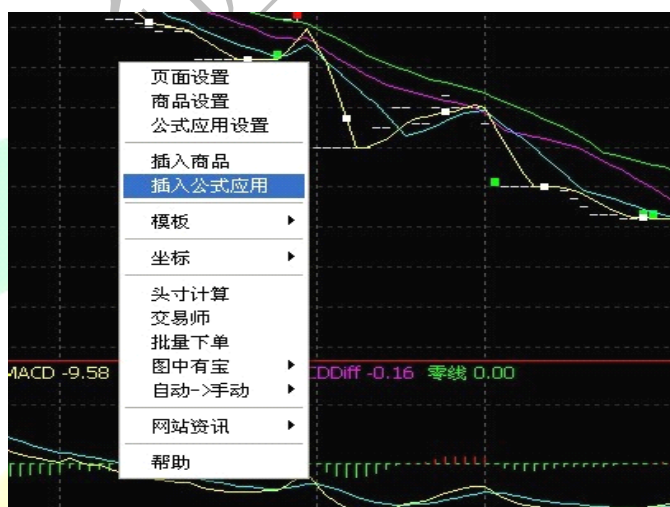


Step 3: 在公式编辑器中, 输入 sample1 的代码;



Step 4: 点击工具栏中的“校验保存公式”（即红圈处）进行代码编译;

Step 5: 在超级图表右键菜单里选择“插入公式应用”;



Step 6: 在弹出的菜单中选择编写好的公式（例 1 公式名保存为 wyj）-->点击调用（黑圈处）



Step 7: 到指定的文件夹查看结果



这就是公式运行的结果

## 2.2 TB 公式的基本知识点

本例中大家会看到，参数和变量在结构上它们是相似的，其形式都是：

数据类型+变量名

它就像是我们中国人的名字一样，数据类型是我们的姓氏，是规定好的；而变量名就像我们的名字，叫大刚叫杨过都行，可以自由发挥。下面我们将介绍 TB 中的数据类型。

### 2.2.1 TB 公式中有三种基本的数据类型

数值型(Numeric)

字符型(String)

布尔型(Bool)

为了对变量、参数进行回溯，又增加了序列类型

数值型序列变量/参数(NumericSeries)

字符型序列变量/参数(StringSeries)

布尔型序列变量/参数(BoolSeries)

为了通过用户函数返回多个值，又增加了引用类型

数值型引用(NumericRef)

字符型引用(StringRef)

布尔型引用(BoolRef)

### 2.2.2 序列变量与普通变量的比较

	序列变量	普通变量
共同点	都是变量、声明方法一样、都可以对其赋予默认值、用法相同	
不同点	可通过"[n]"来回溯以前的变量值	不行
	执行速度相对较慢，占内存多	执行快，占内存少
	在指定条件下对某变量赋值，序列变量的会自动传递上一个Bar的变量值，直至语句对于进行新的赋值	只在条件满足的Bar上会有指定的赋值，其它Bar上则仍会记录

第一章提到的 bar 数据就是这里说到的序列变量数据了。

### 2.2.3 参数的声明和使用

1. 参数在使用前必须进行声明，如：

Params



NumericSeries Price(1);

Numeric Length(10);

括号内数字为初值;参数的值在公式的脚本段中只能引用, 不能修改;

2. 公式应用和用户函数的参数略有不同:

①公式应用的参数只支持三种基本类型, 用户函数的参数支持全部九种类型;

②公式应用的参数一定要有初始值, 而用户函数的参数可以没有默认值;

提示: 这里大家只需对公式应用和用户函数有不同有个初步的印象, 下文我们会针对例子进行讲解, 涉及到的点我们会一一指出, 到时大家再回看这里, 便会豁然开朗。

#### 2.2.4 变量和变量的声明

变量的主要用处在于它可以存放计算或比较的结果, 以方便在之后的脚本中直接引用运算的值, 而无需重现计算过程。

变量使用前必须进行声明, 声明方法如下:

Vars

Numeric AvgValue;

#### 2.2.5 变量的赋值 (变量类型和表达式的类型要一致)

变量名称 = 表达式;

一定是左边的内容赋给右边, 二者不能调换

例如: AvgValue = Summation(Price, Length) / Length;

意思是 Summation(Price, Length) / Length 的结果赋给 AvgValue;

这里 Summation 是一个求和函数。

## Summation

说明	求和
语法	Numeric Summation(NumericSeries Price, Numeric Length)
参数	Price 用于求和的值, 必须是数值型序列值; Length 是需要计算的周期数, 为整数。
备注	该函数计算指定周期内的数值型序列值之和, 返回值为浮点数; 当序列值的 CurrentBar 小于 Length 时, 该函数返回无效值。
示例	Summation(Close, 12); 计算12周期以来的收盘价的和; Summation((Close + High + Low) / 3, 10); 计算10周期以来高低收价格的平均值的和。



## 例二 求平均

实现代码如下：

```
Params    //声明参数
NumericSeries Price(1); //定义价格参数，初始值为 1
Numeric Length(10);
Vars    //声明变量
Numeric AvgValue; //定义变量 AvgValue
Begin
AvgValue = Summation(Price, Length) / Length;
    //和除以时间长度的值赋给变量 AvgValue
Return AvgValue; //此函数返回 AvgValue 的值
End
```

} 参数  
} 变量

注意：每条代码//后的字为注释，方便读程序。程序不会执行//后内容。

本例子是一个用户函数，大家对比公式运用会发现，其实这两者是没有区别的，只是用户函数会用 return 返回结果，而公式运用则没有返回值。例三就是一个公式运用的例子，大家可以进行对比。

## 2.3 运用 TB 公式实现策略

### 例三 双均线系统

交易规则：

1. 当 5 均线上穿 10 均线时平空做多 1 手；
2. 当 5 均线下穿 10 均线时平多做空 1 手；

实现代码如下：

```
Params
Numeric length1(5);
Numeric length2(10);
Numeric lots(1);
Vars
NumericSeries ma5;
NumericSeries ma10;
Begin
ma5=Average(close, length1); //变量的赋值
ma10=Average(close, length2);
if (CrossOver(ma5, ma10)) //如果 5 均线上穿 10 均线这个条件为真
```

} 参数  
} 变量

```
{
    Buy(lots,close);//平空多做一手
}
if (CrossUnder(ma5,ma10))//如果 5 均线下穿 10 均线
{
    Sellshort(lots,close);//平多做空一手
}
PlotNumeric("ma5:",ma5);//输出 5 均线的连线
PlotNumeric("ma10:",ma10);//即调用这个函数后，k 线图上会显示 10 均线
End
```

if 条件语句

### 2.3.1 参数的作用

本例中我们使用的是 5 均线和 10 均线，其实我们可以不用设置参数，代码如下：

```
Vars
NumericSeries ma5;
NumericSeries ma10;
Begin
ma5=Average(close,5);
ma10=Average(close,10);
...
End
```

可是！如果改用 20 天做周期，我们必须改程序，把 5 改成 20，然后编译。下次想用别的周期，还得改，非常麻烦。

如果使用参数，就方便多了。程序写好，使用时改参数就好了。

因此，在本程序中我们还是选择使用参数。如果想用 10 均线和 20 均线作比较，我们直接改参数就好了。代码如下：

```
params
Numeric length1(5);
Numeric length2(20);
numeric lots(1);
```

...

End

实际上，在此程序中，我们就只用改变黄色部分一条语句。省了很多功夫  
如果不想改代码，我们还可以直接在界面上改参数。操作如下：

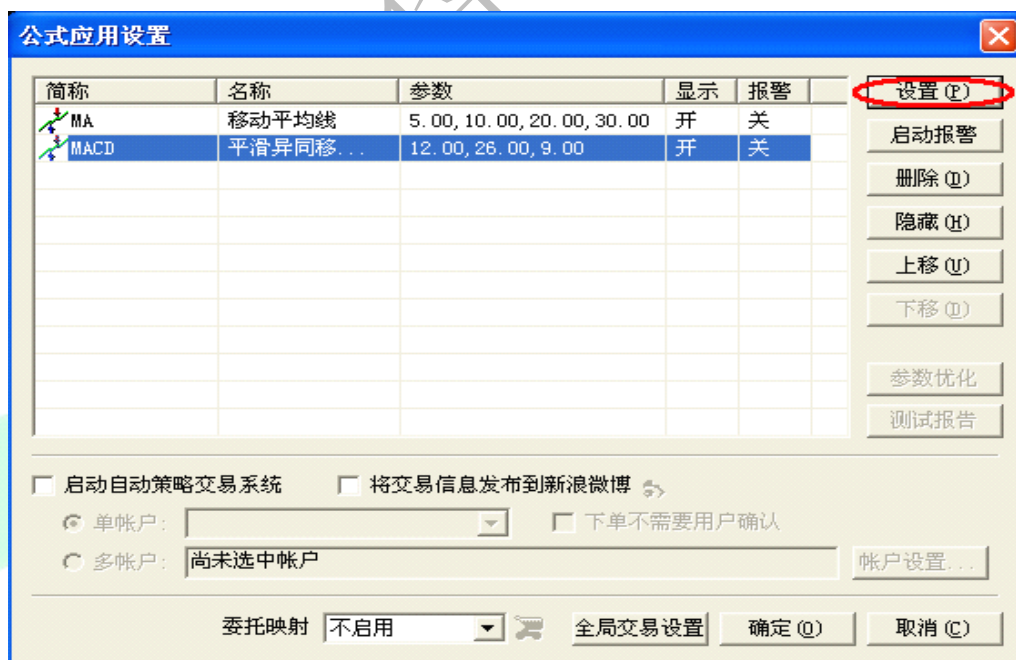
Step 1:

鼠标右键选择公式应用设置



Step 2:

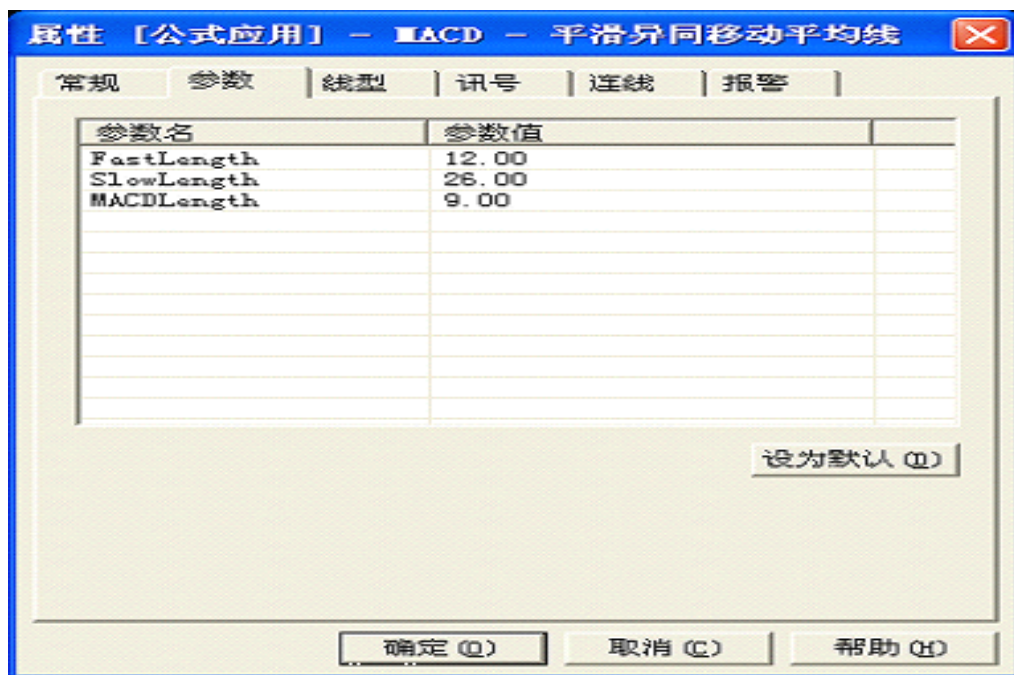
选择要进行参数修改的公式，点击设置



Step 3:

按照需要进行参数值的修改





### 2.3.2 控制语句

#### 1. 条件语句:

if 语句

if - else 语句

if - Else if 语句

if - Else 嵌套

#### 2. 循环语句:

For 循环变量 = 初始值 TO 结束值

For 循环变量 = 初始值 Downto 结束值

While 循环

本例中用到的就是 if 语句我们就比例进行讲解

语法如下:

```
If (条件)
{
    TB 公式语句 1;
}
...
```

①如果条件为真, 执行 {} 内语句; 否则, 跳过 {} 内容, 执行下面的语句。

②如果 TB 公式语句是单条，您可以省略 {}，二条或者二条以上的语句必须使用 {} 我们就此例进行讲解。

if (CrossOver(ma5,ma10))//括号内条件为真，执行 {} 内语句，否则，跳过 {} 执行下面语句

```
{
    Buy(lots,close); //平空多做一手
}
```

//像此例，{} 内语句只有一条，就可以省略 {}

其他控制语句的用法和 if 语句是大同小异的，只要理解好这个，其他都可以套用。

这段代码是一个交易的策略的实现。我们最终的目标是让计算机自动执行策略，因此完成公式的编写之后，我们还要将它加载到 k 线图中。因此，该实验就比实验 1 多一个加载的操作。

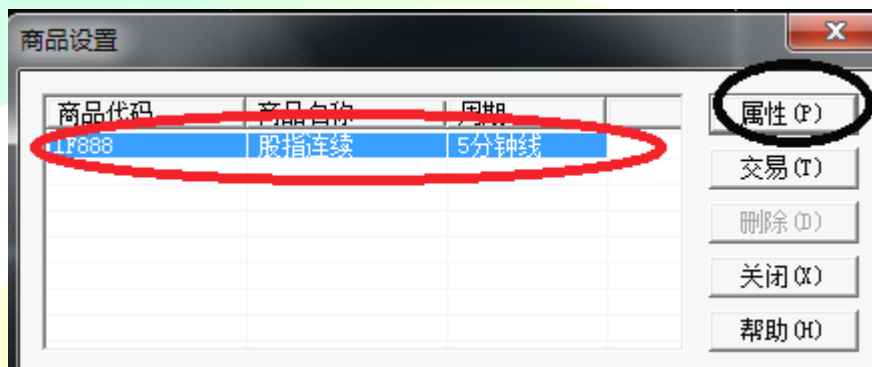
实验步骤：

按实验 1 的步骤执行完 Step 6

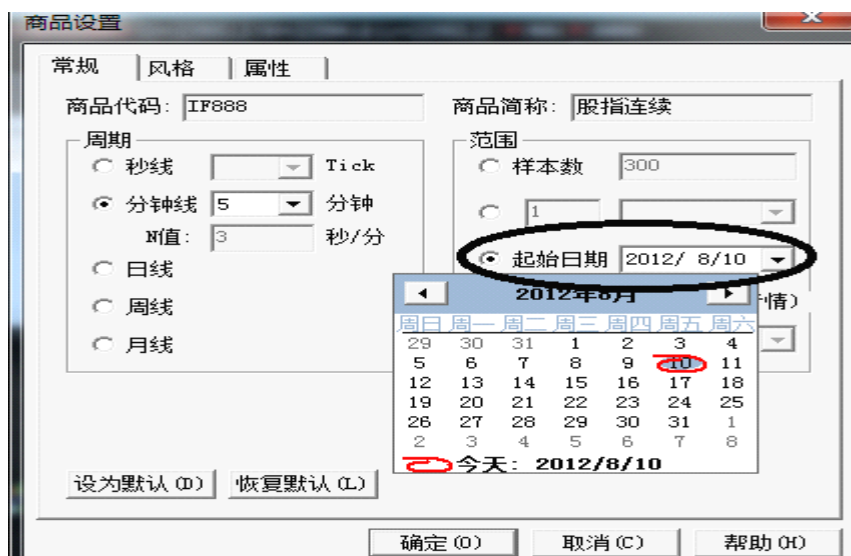
Step7:在超级图表上点击右键→点击商品设置



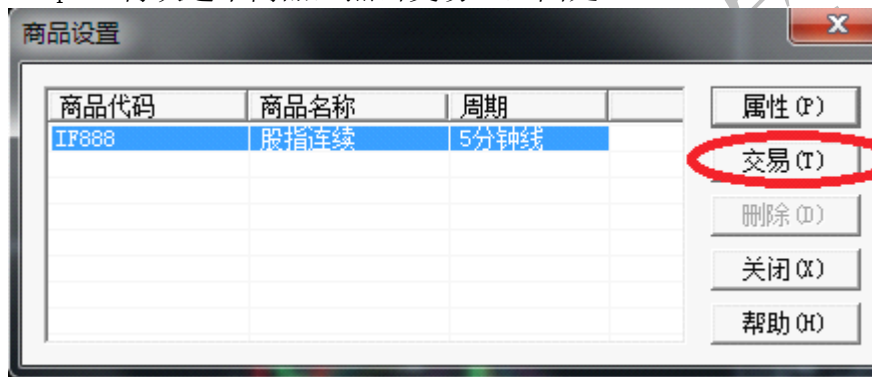
Step8:选中商品（红圈处）→点击属性（黑圈处）



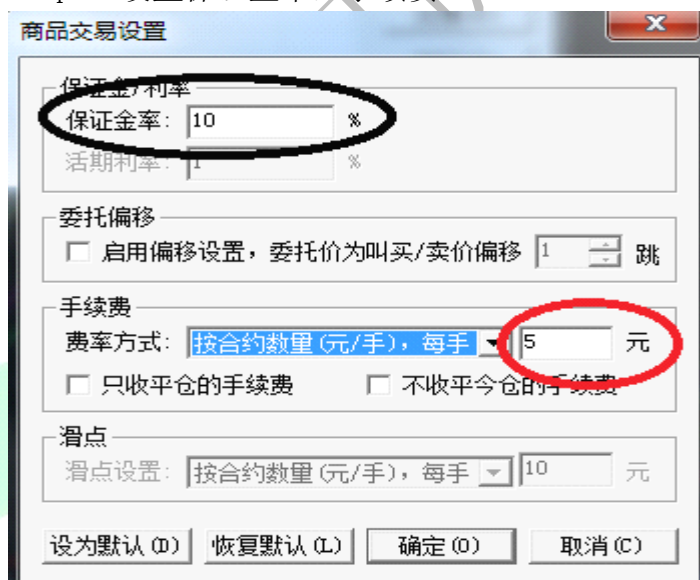
Step9: 设置测试数据的起始日期（点击黑圈处）→下拉时间表设置时间



Step10: 再次选中商品，点击交易（红圈处）



Step11: 设置保证金率，手续费



以上，就是公式从编写到运行的全部流程了。



TB 中默认不能进行同一方向连续建仓（卖和买是相反方向建仓），所以如果您想连续建仓，需要进行全局交易设置，步骤如下：

Step 1:

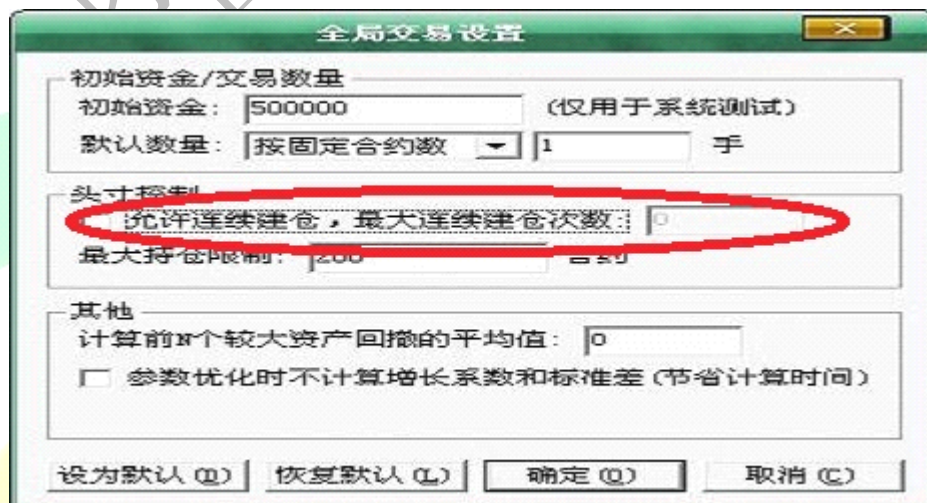
鼠标右键，选择公式应用设置



Step 2: 点击红框中全局设置



Step 3: 进行全局交易设置（红圈处）



这样，全局交易也就设置好了。

注意：

这部分编程的内容是针对历史回测进行讲解的，不包括实盘。无论是历史回测还是实盘编程，流程都是一样的。在后续的材料中，我们将会介绍历史回测与实盘的异同。

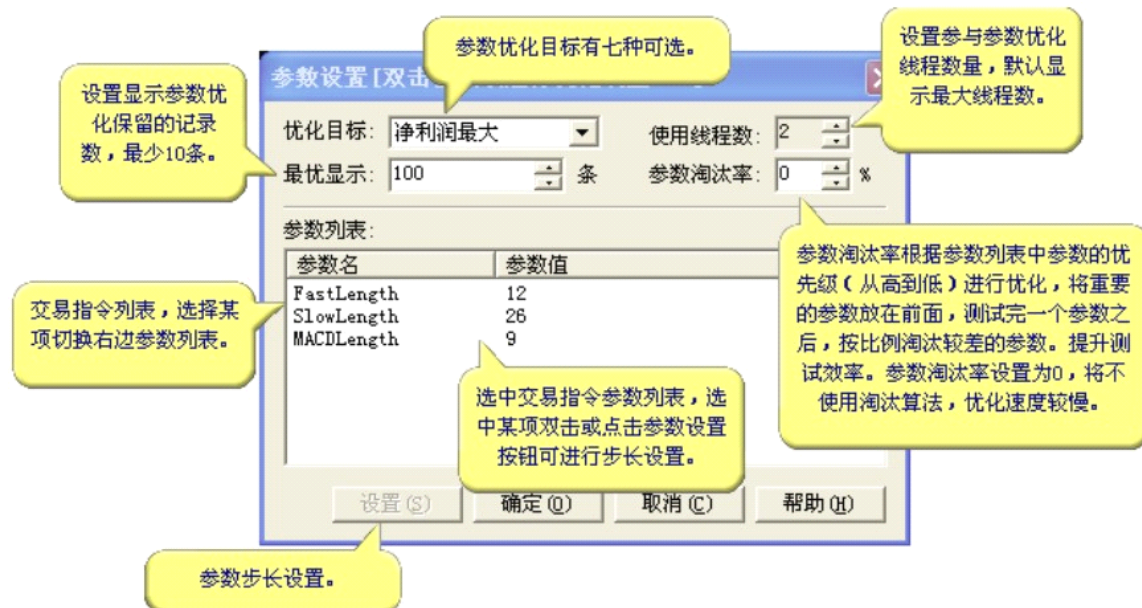
本章小结：

以上就是一个交易策略从编写到加载的流程了。通过例二（用户函数）与例三（公式运用）的对比，你现在可以一一验证在参数那一章节讲到的用户函数与公式运用的不同指出了吧。但是这两者的逻辑结构是一样的。如果您想做到自己编写策略来实现交易，那么接下来要做的，就是熟悉 TB 中库函数的功能和用法。库函数都是现成的，操作中我们就只需要调用，就可以实现不同的策略了。大家也许会疑问：库函数这么多，什么时候才看得完。而实际上，我们只要先熟悉几个常见的函数，就已经可以实现一些基本的小策略。像本资料举的第二个例子，就已经包含了很多常用的函数，详细用法查阅 TB 帮助即可。库函数成千上万，本资料中我们就不一一赘述了。

## 第三章 交易策略参数优化

### 3.1 参数优化基本操作

在超级图表中插入一个或一个以上交易指令后，菜单和工具栏中的交易策略参数优化选项将会有效，您可以通过点击菜单项或工具栏使用该功能。交易策略参数优化模块可对多个交易指令组合的所有参数进行优化，您可以通过参数设置界面对需要优化的参数进行设置，界面如下：



参数优化目标有以下九种选项:

净利润最大: 以优化结果中的净利润最大为目标, 保留指定数量的记录数;

交易手数最大: 以优化结果中的交易手数最大为目标, 保留指定数量的记录数;

平均净利润最大: 以优化结果中的平均净利润最大为目标, 保留指定数量的记录数;

盈利因子最大: 以优化结果中的盈利因子最大为目标, 保留指定数量的记录数;

盈利比率最大: 以优化结果中的盈利比率最大为目标, 保留指定数量的记录数;

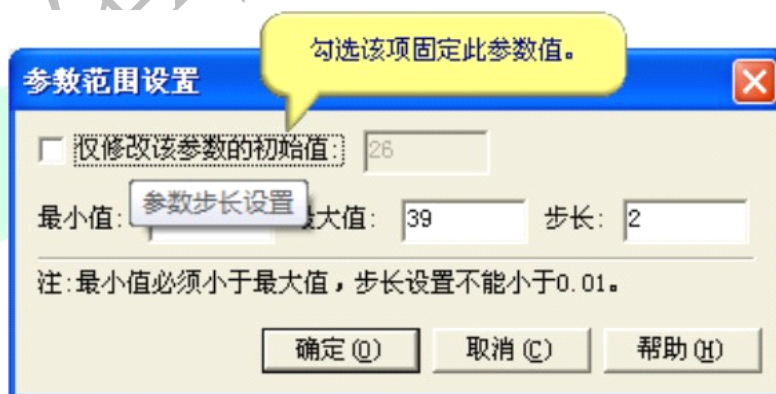
盈亏比率最大: 以优化结果中的盈亏比率最大为目标, 保留指定数量的记录数;

收益率最大: 以优化结果中的收益率最大为目标, 保留指定数量的记录数;

头寸系数最大: 以优化结果中的头寸系数最大为目标, 保留指定数量的记录数;

TB 系数最大: 以优化结果中的 TB 系数最大为目标, 保留指定数量的记录数。

通过双击参数项或点击参数设置按钮, 对选中的参数项进行步长设置, 界面如下:



您可以根据自己需要对各项参数进行点击, 更改到你想要的值, 然后点击确定按



钮就完成了。系统会自动进行参数优化的计算，优化的进程以及有效信息都会有显示，您可以通过点击取消按钮终止计算。

注意：根据优化参数设置的多少，优化时间可能长达几小时甚至数天，优化过程中，CPU 将会大量被占用，您可以选择空闲的时间进行大量参数优化计算。

参数优化计算完成之后，将会显示出最优的记录，您可以通过工具栏的【交易设置】按钮修改交易参数，也可以通过重新优化按钮进行重新计算。

### 3.2 优化结果生成分析图表

参数优化结果会以一个报表的形式呈现：

交易策略参数优化[SR888 - 1日线]

#	P:boLength	P:fsLength	净利润	盈利比率	平均利润	交易手数	最大资产回撤	TB系数	增长系数
1*	22.00	57.00	65360.00	47.54	1071.48	61	-34630.00	14.68	24.07
2	22.00	52.00	65360.00	47.54	1071.48	61	-34630.00	14.68	24.07
3	22.00	47.00	65360.00	47.54	1071.48	61	-34630.00	14.68	24.07
4	14.00	27.00	64220.00	51.35	867.84	74	-34130.00	16.40	22.16
5	22.00	27.00	62180.00	47.95	851.78	73	-34630.00	13.85	18.97
6	30.00	62.00	61990.00	50.00	1192.12	52	-33870.00	20.93	40.26
7	30.00	57.00	61990.00	50.00	1192.12	52	-33870.00	20.93	40.26
8	30.00	52.00	61990.00	50.00	1192.12	52	-33870.00	20.93	40.26
9	30.00	47.00	61990.00	50.00	1192.12	52	-33870.00	20.93	40.26
10	22.00	37.00	60080.00	43.28	896.72	67	-34630.00	11.90	17.76

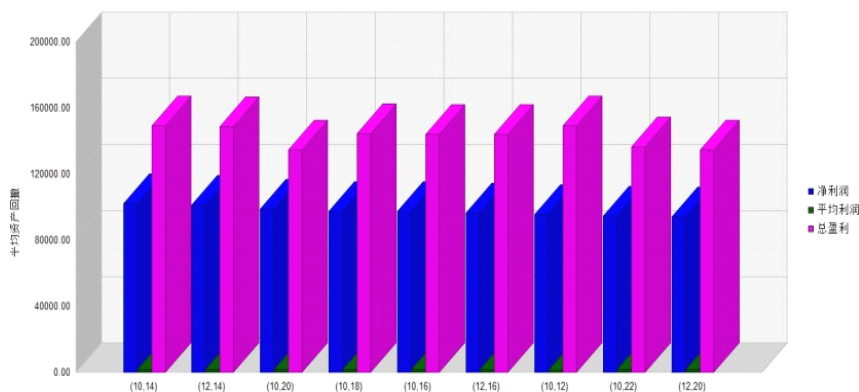
上图为参数优化报表的一部分。您可以选定某字段上的部分数据点生成报表的按钮，有三种不同类型的图表可选择。从而生成如下图的个性化报表图型：

交易策略参数优化[SR888 - 1日线]

#	P:boLength	P:fsLength	净利润	盈利比率	平均利润	交易手数
1*	22.00	57.00	65360.00	47.54	1071.48	61

生成报表

生成如下图：



### 免责声明:

本报告的信息来源于已公开的资料,本公司对该信息的准确性、完整性或可靠性不作任何保证,报告中的内容仅供学习参考。本报告的版权为我公司所有,未经书面许可任何机构和个人不得以任何形式翻版、复制、刊登、发表或者提供他人使用。