

Справочное руководство по языку Calculon

Содержание

1. Введение.....	3
2. Синтаксис.....	4
2.1. Подсветка синтаксиса в функциональном блоке Program.....	5
3. Типы данных.....	7
4. Операторы.....	9
4.1. Арифметические операторы.....	9
4.2. Тригонометрические операторы.....	10
4.3. Операторы сравнения.....	11
4.4. Логические операторы.....	12
4.5. Оператор присваивания.....	13
4.6. Операторы условия.....	13
4.7. Бинарные операторы.....	15
4.8. Приоритет операторов.....	15
4.9. Функция текущего времени.....	16
5. Ссылки на атрибуты функционального блока.....	17
6. Примеры.....	18

1. Введение

Calculon - это язык программирования с С-подобным синтаксисом.

Этот язык предназначен для задач, которые не могут быть решены при помощи стандартной библиотеки функциональных блоков (далее - ФБ), например:

- подстановка математической формулы;
- большое количество условных переходов;
- моделирование объекта управления.

Особенности Calculon:

- динамическая типизация;
- отсутствуют процедуры и рекурсия.



Внимание: Справочная информация доступна:

- из главного меню командой **Помощь > Справка**;
- по клавише **“F1”**;
- выбором пункта **Справка** из контекстного меню дерева проекта.

2. Синтаксис

Программа на Calculon состоит из *выражений*. Выражение, в свою очередь, состоит из *оператора* и его *операнда*.

Оператор - это описание действия над данными. Операнды - это данные, которые программа обрабатывает при помощи оператора.

Операндами выражения могут быть:

- переменные;
- литералы;
- другие выражения.

Переменная - это зарезервированные места в памяти для хранения значений. Переменной присваивается определенный тип данных.

Литерал - фиксированное значение, представленное прямо в коде.

Пример записи выражения:

```
a=b+4;
```

где *a* - переменная-операнд;

b+4 - выражение-операнд;

= - оператор.

Выражения могут объединяться в блоки, отделенные фигурными скобками. Переменные, созданные в блоке, могут быть использованы только внутри него. Ниже представлен фрагмент кода с комментариями:

```
a=2;  
{  
  a=3;  
  b=a;
```

- *b* равно 3

```
}  
c=a;
```

- *c* равно 3

```
d=b;
```

Ошибка: *b* - локальная переменная предшествующего блока.

Пример записи комментария:

```
// comment  
/* comment */
```

Правила синтаксиса:

- любое выражение заканчивается точкой с запятой;
- имена переменных могут состоять из цифр, латинских букв, знаков подчеркивания;
- имена переменных не должны начинаться с цифры;
- Calculon чувствителен к регистру;
- в качестве переменной нельзя использовать ключевые слова.

В Calculon зарезервированы **ключевые слова**:

- if;
- else;
- true;
- false;
- static.

2.1. Подсветка синтаксиса в функциональном блоке Program

Файл со стилями подсветки располагается по следующему адресу:

1. для ОС **Linux**: </opt/Nafta/NaftaProcess/Vision/styles/calculon.css>;
2. для ОС **Windows**: <C:\\Program Files\\NaftaProcess\\Engineer Station\\styles\\calculon.css>.

Ниже приведен перечень пунктов, отвечающих за стиль подсветки:

- Ключевых слов "if", "else", "true", "false", "static", "sin", "cos", "tan", "asin", "acos", "atan", "atan2", "sinh", "cosh", "tanh";

```
.keyword {  
    -fx-fill: purple;  
    -fx-font-weight: bold;  
}
```

- Скобок "("");

```
.paren {  
    -fx-fill: firebrick;  
    -fx-font-weight: bold;  
}
```

- Скобок "{}";

```
.brace {  
    -fx-fill: teal;  
    -fx-font-weight: bold;  
}
```

- Текста в кавычках “**example**”;

```
.string {  
    -fx-fill: blue;  
}
```

- Комментариев;

```
.comment {  
    -fx-fill: cadetblue;  
}
```

- Ссылки.

```
.reference {  
    -fx-fill: darkred;  
    -fx-font-style: italic;  
}
```

3. Типы данных

Каждая переменная в Calculon имеет конкретный тип, от которого зависят размер выделяемой оперативной памяти и применяемые операции для переменной.

Типы переменных в Calculon:

- целое знаковое число (4 байта оперативной памяти);
- вещественное число (4 байта оперативной памяти);
- булевый логический тип (1 байт оперативной памяти);
- число в шестнадцатеричном формате (8 байтов оперативной памяти)¹;
- длинное целое знаковое число (8 байтов оперативной памяти)².

Calculon - язык с динамической типизацией, то есть тип переменной определяется во время присваивания значения:

Таблица 1. Примеры определения типа данных

Тип данных	Пример
целое знаковое число	intValue = -1;
вещественное число	realValue = 22.3;
булевый логический тип	boolValue = false;
шестнадцатеричное число	hexValue = 0xAAAABBBBBCCCCDDDD;
длинное целое знаковое число	long_intValue = 1245678987654321;

Все типы имеют диапазон максимального и минимального значений:

Таблица 2. Диапазон типов данных

Тип данных	Диапазон принимаемых значений
целое знаковое число	-2 147 483 648 / 2 147 483 647
вещественное число	$\pm 1.175494351 \times 10^{-38}$ / $\pm 3.402823466 \times 10^{38}$
булевый логический тип	false / true
шестнадцатеричное число	0x0 / 0xFFFFFFFFFFFFFFFF

¹ Шестнадцатеричное число имеет вид 0xXXXXXXXXXXXXXXXXXX, где 0x - префикс обозначающий формат числа, XXXXXXXXXXXXXXXXXXXX - числа от 1 до 9 и буквы латинского алфавита от aA до fF.

² Необходимо добавить знак "l" после числа без пробела.

Тип данных	Диапазон принимаемых значений
длинное целое знаковое число	-9223372036854775807/9223372036854775807

Явного приведения типов в Calculon нет, оно происходит автоматически. Если в какой-либо операции участвуют переменные разных типов, то все эти переменные приводятся к наибольшему типу:

```
a=1+true;
```

$a = 2$, true приводится к целому числу 1.

```
b=4.5+3;
```

$b = 7.5$, 3 приводится к вещественному числу 3.0.

```
c=3.0+true;
```

$c = 4.0$, true приводится к вещественному числу 1.0.

4. Операторы

В Calculon существуют следующие операторы для управления переменными:

- арифметические операторы;
- операторы сравнения;
- логические операторы;
- оператор присваивания;
- операторы условия;
- бинарные операторы;
- оператор скобок;
- функция текущего времени.

4.1. Арифметические операторы

Арифметические операторы используются в математических выражениях.

Описание и примеры использования арифметических операторов в Calculon представлены в Таблице 2. Описание констант представлено в таблице:

Таблица 3. Примеры использования арифметических операторов

Обозначение оператора	Описание	Пример использования (пусть $a = 30$, $b = 10$)
+	Складывает значения операндов по обе стороны от оператора	$a + b == 40$
-	Вычитает значение правого операнда из значения левого операнда	$a - b == 20$
*	Умножает значения операндов по обе стороны от оператора	$a * b == 300$
/	Делит значение левого операнда на значение правого операнда. Если левый и правый операнды являются целыми числами, то деление осуществляется по правилу целочисленного деления	$a / b == 3$
$a \% b$	Делит по модулю (получает остаток при выполнении целочисленного деления), в том числе и для float	$10 \% 3 == 1$

Обозначение оператора	Описание	Пример использования (пусть $a = 30$, $b = 10$)
$\text{sqrt}(a)$	Извлекает квадратный корень из аргумента, представленного в скобках. Если результат операции - иррациональное число, то его округление регулируется настройками	$\text{sqrt}(25) == 5$
$\ln(x)$	Вычисляет натуральный логарифм x	$\ln(\text{EXP}) == 1$
$\lg(x)$	Вычисляет десятичный логарифм x	$\lg(100) == 2$
$\log(x,y)$	Вычисляет логарифм y по основанию x	$\log(2,32) == 5$
$\text{pow}(x,y)$	Возводит x в степень y	$\text{pow}(2,9) == 512$
$\text{expow}(x)$	Возводит константу e в степень x	$\text{expow}(1) == e$

Описание констант представлено в таблице:

Таблица 4. Константы

Обозначение константы	Описание
PI	Число $\pi = 3.14159265358979323846$
EXP	Основание натурального логарифма $e = 2.7182818284590452354$

4.2. Тригонометрические операторы

Тригонометрические операторы используются в математических выражениях.

Описание и примеры использования тригонометрических операторов в Calculon представлены в таблице:

Таблица 5. Примеры использования тригонометрических операторов

Обозначение оператора	Описание	Пример использования
$\sin$	Вычисляет синус в радианах (rad)	$\sin(a+b) == 0.5984271$ (при $a = 0.5$, $b = 2.0$)

Обозначение оператора	Описание	Пример использования
cos	Вычисляет косинус в радианах (rad)	$\cos(a+b) == -0.801143616$ (при $a = 0.5, b = 2.0$)
tan	Вычисляет тангенс в радианах (rad)	$\tan(a+b) == 0.84228838$ (при $a = 0.5, b = 0.2$)
asin	Вычисляет обратный синус в радианах (rad)	$\text{asin}(a+b) g == 0.77539748$ (при $a = 0.5, b = 0.2$)
acos	Вычисляет обратный косинус в радианах (rad)	$\text{acos}(a+b) == 0.79539883$ (при $a = 0.5, b = 0.2$)
atan	Вычисляет обратный тангенс в радианах (rad)	$\text{atan}(a+b) == 0.610725964$ (при $a = 0.5, b = 0.2$)
atan2	Вычисляет обратный тангенс (на основе x,y) в радианах (rad)	$\text{atan2}(a+b) == 1.19028995$ (при $a = 0.5, b = 0.2$)
sinh	Вычисляет гиперболический синус в радианах (rad)	$\sinh(a+b) == 0.75858370$ (при $a = 0.5, b = 0.2$)
cosh	Вычисляет гиперболический косинус в радианах (rad)	$\cosh(a+b) == 1.25516900$ (при $a = 0.5, b = 0.2$)
tanh	Вычисляет гиперболический тангенс в радианах (rad)	$\tanh(a+b) == 0.60436778$ (при $a = 0.5, b = 0.2$)

4.3. Операторы сравнения

Описание и примеры использования операторов сравнения в Calculon представлены в таблице:

Таблица 6. Примеры использования операторов сравнения

Обозначение оператора	Описание	Пример использования (пусть $a = 30$, $b = 10$)
$==$	Проверяет, равны ли значения двух операндов. Если да, то условие является истинным	$(a == b)$ — условие ложно
$!=$	Проверяет неравенство значений двух операндов. Если значения не равны, то условие является истинным	$(a != b)$ — условие истинно
$>$	Проверяет, является ли значение левого операнда больше, чем значение правого операнда. Если да, то условие является истинным	$(a > b)$ — условие истинно
$<$	Проверяет, является ли значение левого операнда меньше, чем значение правого операнда. Если да, то условие является истинным	$(a < b)$ — условие ложно
$>=$	Проверяет, является ли значение левого операнда больше или равно значению правого операнда. Если да, то условие является истинным	$(a >= b)$ — условие истинно
$<=$	Проверяет, является ли значение левого операнда меньше или равно значению правого операнда. Если да, то условие является истинным	$(a <= b)$ — условие ложно

4.4. Логические операторы

Описание и примеры использования логических операторов в Calculon см. Таблица 7. Примеры использования логических операторов.

Значение истина (true) является эквивалентом числа, отличного от 0. Значение ложь (false) является эквивалентом числа 0.

Таблица 7. Примеры использования логических операторов

Обозначение оператора	Описание	Пример использования (пусть <i>a</i> имеет значение <i>true</i> , <i>b</i> имеет значение <i>false</i>)
&&	Логический оператор «И». Если оба операнда имеют значение истины, то условие является истинным	(<i>a</i> && <i>b</i>) — значение <i>false</i>
	Логический оператор «ИЛИ». Если любой из двух операндов имеет значение истины, то условие является истинным	(<i>a</i> <i>b</i>) — значение <i>true</i>
!	Логический оператор «НЕ». Меняет логическое состояние. Если условие имеет значение истины, то оператор логического «НЕ» изменит его на ложь	!(<i>a</i> && <i>b</i>) — значение <i>true</i>

4.5. Оператор присваивания

Оператор присваивания обозначается символом = и присваивает значение правой части выражения к значению операнда из левой части. Например, *a* присвоено значение 2:

```
a=2;
```

Оператор присваивания выполняет роль объявления переменной, если она появляется в программном коде впервые.

Если переменная объявляется как *static*, то она сохраняет свое значение между циклами исполнения технологической программы. (См. пример 1 п. 6. [Примеры](#))

4.6. Операторы условия

В Calculon существуют два оператора условия:

- *if*;
- *if else*.

Оператор if: Если заданное условие является истинным, то выполняется выражение или блок кода.

Синтаксис для оператора if представляется в виде:

```
if (условие) выражение;  
if (условие) {блок кода};
```

Пример:

```
if (a==10) a=0;
```

Если a равно 10, то a присваивается значение 0.

```
if (a==10)  
{  
    a=0;  
    b=10;  
}
```

Если a равно 10, то a присваивается значение 0, b присваивается значение 10.

Оператор if else: Если заданное условие является истинным, то выполняется выражение 1 или блок кода 1. Если заданное условие является ложным, то выполняется выражение 2 или блок кода 2.

Синтаксис для оператора if else представляется в виде:

```
if (условие) выражение else выражение;  
if (условие) {блок кода} else {блок кода};
```

Пример:

```
if (a==10) a=0 else a=a+1;
```

Если a равно 10, то a присваивается значение 0, если a не равно 10, то a присваивается значение $a+1$.

```
if (a==10)  
{  
    a=0;  
    b=10;  
}  
else  
{  
    a=a+1;  
    b=0;  
}
```

Если a равно 10, то a присваивается значение 0, b присваивается значение 10. Если a не равно 10, то a присваивается значение $a+1$, b присваивается значение 0.

4.7. Бинарные операторы

Описание и примеры использования бинарных операторов в Calculon представлены в таблице:

Таблица 8. Примеры использования бинарных операторов

Обозначение оператора	Описание	Пример использования
$x \& y$	Поразрядное «И»	$1FC \& 2D == 8C$
$x y$	Поразрядное «ИЛИ»	$1FC 2D == 1FD$
$x \wedge y$	Поразрядное «ИСКЛЮЧАЮЩЕЕ ИЛИ»	$1FC \wedge 2D == 1D1$
$x \ll y$	Поразрядный сдвиг влево	$1111ABCD \ll 2 == 4446AF34$
$x \gg y$	Поразрядный сдвиг вправо	$1111ABCD \gg 3 == A2223579$
$\sim x$	Поразрядное «НЕ» (инверсия)	$\sim 1111ABCD == EEEE5432$

4.8. Приоритет операторов

Приоритет операторов Calculon определяет порядок их выполнения в выражении. В таблице категории операторов расположены в порядке убывания их приоритета:

Таблица 9. Приоритет операторов

№ п/п	Оператор	Описание
1	$! \sim$	Логическое «НЕ», поразрядное «НЕ»
2	$/ * \%$	Умножение, деление и остаток
3	$+ -$	Сложение и вычитание
4	$\ll \gg$	Побитовое смещение влево и вправо
5	$> < >= <= < > == !=$	Больше, меньше, больше или равно, меньше или равно, равно, не равно

№ п/п	Оператор	Описание
6	& ^	Поразрядное «И», поразрядное «ИСКЛЮЧАЮЩЕЕ ИЛИ», поразрядное «ИЛИ»
7	&&	Логическое «И», логическое «ИЛИ»
8	=	Оператор присваивания

Оператор скобок обозначается символом () и изменяет приоритет операторов внутри выражения:

```
a=2+1*4;
```

Переменной *a* будет присвоено значение 6 согласно приоритету операторов.

```
a=(2+1)*4;
```

Переменной *a* будет присвоено значение 12, т.к. приоритет выполнения переходит к оператору сложения внутри скобок.

4.9. Функция текущего времени

Встроенная функция получения текущего времени `get_time()` — число миллисекунд, прошедших с момента времени 00 часов 00 минут 00 секунд 0000 миллисекунд 01.01.1970 года.

Пример:

```
a = get_time();

/* Get the number of seconds since 1970 */
a = a / 1000;
```

5. Ссылки на атрибуты функционального блока

В Calculon имеется возможность использования в качестве переменной ссылки на атрибут ФБ.

Ссылка оформляется следующим образом: ``имя ФБ.атрибут ФБ`` ("`" – гравис), где атрибут ФБ - значение, которое требуется получить (могут использоваться только атрибуты типов REAL, DINT, BOOL).

Ниже представлены фрагменты кода с комментариями:

Пример 1

```
a = `МК513016AI.ModulePosition`
```

- в переменную *a* будет записан атрибут "Позиция модуля в корзине ФБ МК513016AI".

Пример 2

```
b = ln(`МК513016AI.Output1`)
```

- в переменную *b* будет записан натуральный логарифм атрибута "Выходное значение первого канала в процентах".

6. Примеры

Пример 1

Генератор пилообразного сигнала. Исходные данные: Задан сигнал с амплитудой 100, произвольным периодом и начальным уровнем 0. Старт генератора производится в нисходящей фазе.

```
max_value = 100;
min_value = -100;
static current_value = 0;
static direction = false;

if (direction) {
    current_value = current_value+1;
} else {
    current_value = current_value-1;
}

if (current_value <= min_value) {
    direction = true;
}

if (current_value >= max_value) {
    direction = false;
}
RO1 = current_value;
```

Вычисление периода сигнала. В соответствии с алгоритмом работы выходной сигнал генератора (атрибут RO1) обновляется каждый рабочий цикл контроллера. Согласно спецификации рабочий цикл контроллера составляет 100 мс. Соответственно, период T с учетом исходных данных составит:

$$T = (4 * A * 100) / 1000 = 40\,000 \text{ мс} = 40 \text{ с},$$

где $A = 100$ - заданная амплитуда сигнала.

Пример 2

Кривая разгона апериодического (инерционного) звена первого порядка. Исходные данные: RI1 - значение входного сигнала, RO1 - значение выходного сигнала. Начальное значение выходной величины равно 0. Коэффициент усиления звена k равен 0,01.

```
input_value = RI1;
static last_output_value = 0.0;
k = 0.01;

output_value = k*input_value + (1-k)*last_output_value;
last_output_value = output_value;

RO1 = output_value;
```