

# NetCalculator v1.3

© Podobashev Dmitry / BEOWOLF, 2010

## Краткое описание.

Программа-калькулятор, написан на C# под .Net Framework 2.0. Отличительной его особенностью является компактный интерфейс, и возможность вычисления выражений, записанных в обычной математической форме, с широким набором функций. Анализ выражений базируется на рекурсивном нисходящем парсере.

## Возможности программы.

- Вычисление математических выражений, содержащих операции, скобки, константы и функции.
- Работа с десятичными и шестнадцатеричными числами.
- Возможность вычисления нескольких выражений сразу, перечисленных через запятую.
- Широкий набор встроенных математических функций. Есть функции с переменным количеством аргументов, а также функции с аргументами по умолчанию.
- Возможность дополнения пользовательскими переменными и функциями.
- Проверка синтаксиса выражений по мере их ввода.
- Автодополнение вводимых встроенных функций, с информацией по ним.
- Удобный компактный интерфейс. Программа может иметь различную прозрачность в активном и неактивном режиме, также может быть поверх других окон. Шрифт масштабируется с изменением размеров окна.

## Интерфейс.

Основное окно разделено сплитером на 2 части. Левая часть служит для написания выражений для вычисления, в правой части отображается результат. Для вычисления выражения нужно нажать Enter, или выбрать пункт Evaluate в контекстном меню левой части. Ниже этого пункта в контекстном меню есть команды для стандартных операций с буфером обмена Windows, с использованием текста с форматированием.

Во время набора и редактирования выражений, при наборе имени какой-либо функции, под основным окном будет отображаться всплывающий список со всеми встроенными функциями и константами, которые могут быть в выражении. При начале набора имени функции, совпадение будет отмечаться в списке, и можно нажатием на Enter завершить ввод имени функции - сразу вставить нужное имя функции. Например, нажав на «s» в списке отобразится «sin», после чего нажатием на Enter он появится в поле для ввода выражения. В соседних колонках списка показан список ожидаемых аргументов функции, комментарии (отображают метод вычисления функции), и синонимы (их можно писать вместо данного названия). Чтобы отобразить список в любой момент во время набора текста, нужно нажать на Ctrl+J, либо выбрать пункт Built-in lexemes контекстного меню. Если выбрать в списке другое имя функции, то набранные символы будут заменены выбранным именем функции. Чтобы скрыть список, можно нажать на Esc. Нажатие Esc или при скрытом списке очистит содержимое левой и правой части.

Во время редактирования выражения справа будет отображаться значок с восклицательным знаком, если выражение содержит ошибки и не может быть корректно

вычислено. При наведении мыши на значок отобразится подсказка о сути ошибки. При попытке вычисления, место с ошибкой будет выделено. Чтобы отключить «горячую проверку», снимите галку с пункта меню Hot checking. А если будет помечен пункт Hot calculating, то результат вычисления выражения будет отображаться в правой части по мере набора текста.

Если помечен пункт меню Scale by zooming, то при изменении размеров окна будет масштабироваться также шрифт пропорционально высоте окна. Если пункт не помечен, то размер шрифта будет не масштабироваться, а численно меняться, разница состоит в том, что при копировании текста с помеченным пунктом, размер шрифта будет оставаться постоянным, равным 16, а если пункт не помечен, то размер шрифта будет таким «как есть». Тип шрифта можно определить пунктом ниже.

В контекстном меню левой части можно также установить процент прозрачности в активном и неактивном состоянии окна, режим «всегда сверху» и полную прозрачность фона в неактивном состоянии (если помечен пункт меню Inactive transparency).

Редактируемое выражение может содержать целые шестнадцатеричные числа, которые обязательно должны начинаться с префикса «0x». Чтобы отображать результат вычисления в шестнадцатеричной форме, нужно пометить пункт Hexadecimal контекстного меню правой части.

## Синтаксис выражений

Регистр букв в выражении различается, кроме регистра букв шестнадцатеричных чисел.

**Десятичные числа.** Выражения могут содержать десятичные числа с точкой и мантиссой, записанные в принятом формате языка Fortran. Примеры правильного написания чисел: 100, 3,1416, 1,024e3, 1,23e-2. Вместо десятичной запятой можно писать точку. Перед числами и операндами можно ставить унарный плюс или минус.

**Шестнадцатеричные числа.** Шестнадцатеричные числа должны состоять из десятичных цифр и букв от «A» до «F» включительно, без учёта регистра. Перед началом шестнадцатеричного числа нужно написать префикс «0x». Примеры шестнадцатеричных чисел: 0x0d, 0x0A, 0X64, 0xFFFF.

**Операторы.** Между операндами должны стоять операторы, которые делятся на группы по приоритету выполнения. Высшим приоритетом обладают скобки, выражение в которых вычисляется в первую очередь, низшим приоритетом обладает перечисление, выполняемое в последнюю очередь. Из арифметических операций высшим приоритетом обладает возведение в степень, ниже - мультипликативные операции, ниже - аддитивные. Далее приведён список допустимых операторов в порядке возрастания приоритета операции.

➤ Перечисление:

«,» синоним «comma». Перечисление используется для разделения логически независимых элементов выражений, например аргументов функций. Если перечисляются числа, то после запятой перечисления должны стоять пробелы, чтобы эти запятые не были интерпретированы как десятичные запятые.

➤ Булевские операции:

«&&» синоним «and». Логическая конъюнкция, логическое «и». Результат операции 1, только если оба операнда не равны 0, иначе результат 0.

«||» синоним «or». Логическая дизъюнкция, логическое «или». Результат операции 1, если хотя бы один из операндов не равен 0, иначе результат 0.

«@@» синоним «xor». Логическая разделительная конъюнкция, сумма по модулю 2, логическое разделительное «или». Результат операции 1, только если только один из операндов не равен 0, иначе результат 0.

«->» синоним «imp». Логическое следование, импликация. Результат операции 0, только если из верного утверждения следует неверное, то есть если первый операнд не равен 0, а второй равен 0.

«<->» синоним «equ». Эквивалентность, тождественность. Результат операции 1, если оба операнда не равны 0, либо оба операнда равны 0.

➤ Операции сравнения:

Результат операции сравнения равен 1, если условие выполнено, и 0 в противном случае.

«<=». Меньше либо равно.

«>=». Больше либо равно.

«<». Больше.

«>». Меньше.

«==». Равно.

«!=» синоним «<>». Не равно.

➤ Бинарные операции:

«&». Бинарная конъюнкция, побитовое «и».

«|». Бинарная дизъюнкция, побитовое «или».

«@». Бинарная разделительная конъюнкция, сумма по модулю 2, побитовое разделительное «или».

➤ Аддитивные операции:

«+» синоним «plus». Сложение.

«-» синоним «minus». Вычитание.

➤ Мультипликативные операции:

«\*». Умножение.

«/». Деление.

«%» синоним «mod». Остаток от деления.

➤ Показательные операции:

«^». Возведение в степень.

➤ Унарные операции:

«!». Префиксная операция - логическое отрицание. Возвращает 1, если операнд нулевой, и 0, если операнд не нулевой.

«!». Постфиксная операция - факториал. Возвращает факториал операнда, то есть произведение всех натуральных чисел от 1 до значения операнда включительно, либо 1 если операнд - нулевой. Для вещественных положительных чисел с дробной частью, факториал рассчитывается по формуле Стирлинга.

➤ Скобки:

«(» и «)», «[» и «]», «{» и «}». Скобки могут быть круглыми, квадратными и фигурными, и обладают абсолютным высшим приоритетом операций. Тип закрывающей скобки должен соответствовать типу открывающей.

**Константы.** Вместо общеизвестных констант в числовом виде можно использовать их имена. Например, вместо записи числа  $\pi = 3,1415926...$  можно писать **pi**. Перечень допустимых имён есть в контекстном списке, который можно отобразить нажатием Ctrl+J.

**Функции.** Есть ряд встроенных общеизвестных математических функций, список которых можно отобразить нажатием Ctrl+J. Эти функции могут принимать один аргумент, либо несколько аргументов, что также указано в списке. Если функция принимает несколько аргументов, то эти аргументы должны быть перечислены в скобках сразу после имени функции. Например, **sqrt(8, 3)** или **hypot[3, 4]**. Если функция принимает только один аргумент, то в случае, если выражение-аргумент состоит только из одного операнда, скобки можно не писать. Примеры: **ln 10**, **lg exp10**, **hypot{3, 4, sqrt 24}**.

**Перечисления.** Синтаксические правила данного калькулятора позволяют записывать перечисления выражений через запятую. Если перечисление находится на корневом уровне выражения, то все результаты вычисления будут отображены в правой части окна, также через запятую. Перечисления могут быть в скобках, и если это не аргументы функции, то в качестве результата вычисления содержимого скобок будет использовано только первое выражение перечисления.

Примеры выражений.

Выражение	Результат
$(1+2)*3$	9
$(\sqrt{5} + 1)/2$	1,61803398874989
$1,77 / (2*\sin(\pi/4))$	1,25157900270019

Выражение	Результат
$2\pi^{2.25}, \pi^{2.25^2}$	14,1371669411541, 15,9043128087983
$(2\pi^{2.25}, \pi^{2.25^2})$	14,1371669411541
$\sqrt{2\pi^{10}} \cdot (10/e)^{10} / 10!$	0,991704039556062
$\text{hypot}(3, 4), \text{hypot}\{3, 4, \sqrt{24}\}$	5, 7
$\text{deviation}(1.99, 2.27, 2.68, 2.41, 2.52, 2.77, 2.34, 2.09, 2.47, 2.44, 2.39, 2.15, 2.24, 2.02)$	0,224494897901207
$\text{Amean}(1.99, 2.27, 2.68, 2.41, 2.52, 2.77, 2.34, 2.09, 2.47, 2.44, 2.39, 2.15, 2.24, 2.02)$	2,34142857142857
$(\sqrt{5} + 1)/2 == 2\cos(\pi/5)$	1
$\lg \pi > -\ln C$	0
$\sqrt{3} / 2 > \sin(1) \text{ imp } \pi/3 < 1$	0
0x3FFC & 43690	0x2AA8

## Переменные и функции, определённые пользователем.

Всплывающий список, который можно вызвать через Ctrl+J, содержит 3 вкладки. Содержимое первой, список встроенных констант и функций - описан выше, также есть ещё две вкладки для редактируемых переменных и функций.

### Переменные.

Во второй вкладке можно определять переменные, которые в дальнейшем могут участвовать в вычислении выражений. Это реализовано через таблицу из двух столбцов, в левом столбце задаётся имя переменной, в правом - её значение. Каждая переменная интерпретируется каким-либо числом.

Имя переменной должно начинаться с латинской буквы, либо знака подчёркивания, остальные символы имени могут быть также цифрами. Никаких других символов кроме латинских букв, цифр и знака подчёркивания в имени - не допускается. Длина имени может быть произвольной. Имена переменных при парсинге выражений просматриваются сверху вниз. Допускается совпадение имён переменных, в этом случае в выражении участвует значение самой верхней переменной из совпадающих имён. Это может быть полезно при программном рекурсивном помещении переменных в стек, и в других случаях.

Значение переменной определяется выражением в правом столбце. Выражение может содержать вызовы любых встроенных функций и констант, а также другие определённые пользователем переменные и функции. Выражение вычисляется только один раз по окончании его редактирования, после чего в других выражениях в качестве значения переменной участвует число - результат вычисления.

### Функции.

В третьей вкладке определяются пользовательские функции, контекст которых будет интерпретироваться при каждом нахождении в выражении, для переданных значений переменных. Функции задаются таблицей из трёх столбцов - имени функции, перечисления аргументов функции, и тела функции.

Синтаксис имени функции полностью идентичен синтаксису имён переменных. Для функций также допускается совпадение имён, и при совпадении в выражении также будет участвовать контекст самой верхней функции.

После имени функции в следующей колонке должен следовать список аргументов функции. Все аргументы функции должны быть перечислены через запятую. Синтаксис имён аргументов функции идентичен синтаксису переменных. Количество аргументов функции может быть произвольным. При вычислении значения функции в выражении, сначала вычисляются значения аргументов, затем в стек переменных помещаются аргументы функции и соответствующие им значения, в порядке следования в определении. Если имя аргумента функции совпадает с именем определённой раньше переменной, то поскольку аргумент находится вверху стека, то в выражении будет участвовать именно он, а не значение одноимённой переменной.

Функции могут не содержать ни одного аргумента, в этом случае содержимое средней ячейки должно оставаться пустым. Если у функции нет аргументов, то их не следует указывать при вызовах функции, или писать пустые скобки - в этом случае вызов функции синтаксически идентичен указанию переменной. Отличие состоит в том, что при парсинге каждый раз будет вычисляться тело функции, которое, как и для остальных функций, может содержать любые переменные и вызовы функций.

В третьей колонке содержится определение функции. Функция определяется выражением, содержащим перечисленные прежде аргументы.

## Поддержка.

По всем вопросам и предложениям по поводу программы можно связаться по ICQ [320329575](https://www.icq.com/contacts/details.aspx?id=320329575), либо написать письмо на [beowolf@sevstar.net.ua](mailto:beowolf@sevstar.net.ua).

© BEOWOLF, 2010.