

NetCalculator

Этот калькулятор написан на C# под .Net Framework 2.0. Отличительной его особенностью является компактный прогрессивный интерфейс, и возможность вычисления выражений, записанных в обычной математической форме.

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 находятся команды для стандартных операций с буфером обмена Windows, с использованием текста с форматированием.

Во время набора и редактирования текста 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 числом 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 «А» 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.
➤ 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 $\pi = 3,1415926\dots$ 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. Например, `ln 10`, `lg exp10`, `hypot{3, 4, sqrt 24}`.

Поддержка

По всем вопросам и предложениям по поводу программы можно связаться по ICQ [320-329-575](tel:320-329-575), либо написать письмо на beowolf@sevstar.net.ua.

© BEOWOLF, 2007.