

# Verdaqui C Smartcalc Documentation

Калькулятор содержит четыре вкладки:

- 1. Графический калькулятор (основная)
- 2. Классический калькулятор
- 3. Кредитный калькулятор
- 4. Депозитный калькулятор

Программа содержит селектор вкладок в левом верхнем углу.

## Графический калькулятор

Строит графики выражений, записанных в блок `Expressions`.

Чтобы добавить новую строку для выражения, необходимо нажать на кнопку с иконкой `+` (плюс).

После этого можно вписывать математические выражения текстом и они автоматически будут отображены

Пример: `y = sin x` строит график в виде синусоиды.

Есть три типа поддерживаемых выражений - переменные, функции и графики. Также поддерживаются два типа данных - числа и вектора.

Кнопка `Home` с иконкой дома переносит камеру в изначальную позицию.

### Переменные

Задаются в форме: `<var_name> = <expr>`

Например, выражение `c = 2 * e + sin pi` создаст переменную с именем `c` и значением `2 * e + sin pi`. Имя переменной должно быть незанятым, иначе выражение не будет распознано как переменная (а как график)

Если выражение является константным, то есть не зависит от координат графика (`x`, `y`) то значение будет вычислено сразу и отображено снизу.

Если значение не является константным, то есть зависит от `x` и/или `y`, то оно не будет вычислено, но его можно будет использовать в выражениях графиков, словно "функцию без аргументов".

Например, выражения `a = 2x` и `y = a`, если добавлены друг после друга, будут распознаны как переменная и график соответственно. График второго выражения будет эквивалентен графику `y = 2x`.

### Функции

Задаются в форме `<fn_name>(a, b, c, ...) = <expr>`. Имя функции должно быть незанятым. Имена аргументов могут быть любыми. Выражение внутри может использовать аргументы как переменные.

Функция может быть константной и неконстантной. Константная функция - это функция, которая не использует координаты графика `x` и `y`. Неконстантная функция - функция, которая использует координаты.

Константные функции подписываются как `Const function`. Их значения могут быть вычислены без построения графика, если известны входные аргументы.

Пример:

Выражение	Подпись
<code>w(x) = x * e^x</code>	Const function
<code>w(2)</code>	14.78

В данном случае, функция `w(x)` была распознана как константная функция и ее значение было вычислено сразу.

Стоит заметить, что хотя аргумент и называется `x`, но аргумент функции "затеняет" другие переменные с таким названием уровнем выше. Поэтому этот `x` не имеет отношения к координате `x` на графике. Если бы функция была записана так: `w(a) = x * e^x`, то это бы уже не была константная функция.

Пример:

Выражение	Подпись
<code>s(a) = a * sin(x)</code>	Function
<code>y = s(5)</code>	Plot

В данном случае `s(a)` - функция синусоиды с заданной амплитудой. Выражение `y = s(5)` использует эту функцию и строит график, эквивалентный выражению `y = 5 * sin(x)`

### График

Любое выражение, что не подходит под формы `переменной` и `функции` считается `графиком` и будет визуализировано, если не содержит ошибок.

Вы можете менять цвет графика такого выражения в выпадающем меню со значком `V` (треугольник острым концом вниз).

При построении графика, значение выражения вычисляется для каждого пикселя, после чего оно "обрезается" до диапазона от 0 до 1 и преобразуется в цвет. 0 - полностью прозрачный пиксель, 1 - полностью непрозрачный пиксель.

Таким образом, выражение `'0'` никак не изменит изображения, выражение `'0.5'` закрасит полупрозрачным красным (по умолчанию) цветом всю область, а `'1'` закрасит все непрозрачным красным.

Введя, например, выражение `y = sin x` вы получите линию синусоиды (красный цвет в пикселях, которые пересекают линию `y = sin x`).

Доступные значения - числа (`1`, `5.5`, `3e5`) и вектора (`[1, 2, 3]`). Вектора, по аналогии со многими языками программирования, могут быть проиндексированы так: `vector[index]`.

Доступные функции: `sin`, `cos`, `tan`, `asin`, `acos`, `atan`, `ln`, `log`, `slice`, `join`, `min`, `max`.

- `slice` - принимает на вход два вектора (данные и индексы) и возвращает массив с данными из первого элемента, но в порядке (индексов), указанном во втором. Например: `slice([1, 2, 3], [0, 0, 1, 2, 1])` равняется `[1, 1, 2, 3, 2]`.
- `join` - "склеивает" несколько массивов. Пример: `join([1, 2, 3], [4, 5, 6])` = `[1, 2, 3, 4, 5, 6]`
- `min` - находит минимальное значение среди аргументов
- `max` - находит максимальное значение среди аргументов

Важное замечание!! Выражения-графики сильно ограничены, поскольку основаны на шейдерах. В них нельзя использовать неконстантные вычисления, основанные на векторах и не все функции доступны. Вы можете вычислять "важные" данные независимо от графиков и использовать их константную часть в графиках.

Доступные операции:

Операция	Описание
<code>a + b</code>	Сложение
<code>a - b</code>	Вычитание
<code>a * b</code>	Умножение
<code>a / b</code>	Деление
<code>a % b</code>	Остаток от деления
<code>a mod b</code>	Остаток от деления
<code>a ^ b</code>	Возведение в степень
<code>a[b]</code>	Индексация вектора
<code>a = b</code>	Равенство
<code>a != b</code>	Неравенство
<code>a &gt; b</code>	Сравнение (больше)
<code>a &lt; b</code>	Сравнение (меньше)
<code>a &gt;= b</code>	Сравнение (больше или равно)
<code>a &lt;= b</code>	Сравнение (меньше или равно)
<code>a..b</code>	Ряд (вектор от <code>a</code> до <code>b</code> НЕ включительно)
<code>a..=b</code>	Ряд (вектор от <code>a</code> до <code>b</code> включительно)

### Классический калькулятор

Классический калькулятор работает похожим образом на графический калькулятор, однако имеет только одно поле для ввода выражения и не позволяет добавлять функции и переменные.

Он позволяет задать значения для `x` и `y`, после чего вычислить выражение для этих конкретных значений.

### Кредитный калькулятор

Принимает параметры:

- Тип кредита (Аннуитетный/Дифференцированный)
- Сумму кредита (в денежных единицах)
- Процентную ставку (месячную)
- Длительность кредита (в месяцах или в годах)

Вычисляет:

- Месячный платеж
- Общую сумму платежей по кредиту
- Переплату (проценты)

## Депозитный калькулятор

Принимает параметры:

- Сумму кредита (в денежных единицах)
- Процентную ставку (месячную)
- Налоговую ставку
- Длительность (в месяцах или годах)
- Капитализацию процентов (да/нет)
- Период капитализации (в месяцах)
- Список зачислений
- Список снятий

Вычисляет:

- Сумму налогов
- Оставшуюся сумму на депозите
- Общую сумму
- Эффективную процентную ставку