

Функция и оператор Mathcad

В математике есть понятие "*соотношение*": имеются, например, два множества, при этом каждый элемент первого множества *соотносят* с элементом второго множества. Можно сказать и так: любое количество элементов первого множества (любое количество первой группы множеств) соотносят с любым количеством элементов второго множества (любым количеством второй группы множеств). Частным случаем такого соотношения является *функция одного аргумента* $y(x)$ — каждому значению, например, угла (x — первое множество, аргумент функции) соответствует значение его синуса (y — второе множество, первая функция). Читатель средних лет тут же вспомнит знаменитые таблицы Брадиса с колонками таких множеств углов, синусов, логарифмов и других нужных для расчетов величин. Mathcad, как, впрочем, и другие программы численной математики, не хранит множество углов, множество соответствующих им синусов и пр., а рассчитывает эту тригонометрическую функцию по определенному алгоритму, встроенному в Mathcad. Насколько он точен и быстр — это второй вопрос. Другой пример. Есть два множества чисел, одно множество функций, еще одно множество переменных и множество значений определенного интеграла. При этом каждой четверке элементов первых четырех множеств соответствует один элемент пятого множества. Здесь, как понимает читатель, речь идет об операторе определенного интеграла с четырьмя операндами, который также встроен в Mathcad (рис. 1.33).

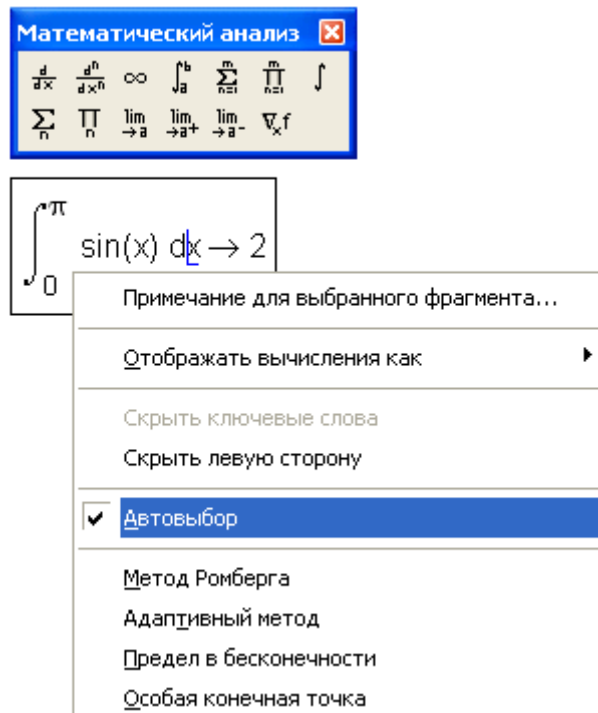


Рис. 1.33. Панель ввода и настройки встроенного оператора "определенный интеграл"

Лет 30 назад наметилось некое несоответствие в трактовке математиками и программистами терминов "функция" и "оператор". Более того, в самом программировании нет полного единства в этом вопросе. То, что мы называем в среде Mathcad оператором, в среде, например, языка BASIC называется функцией. Другой пример. В языке BASIC есть функция $\text{Swap}(a, b)$, меняющая в переменных a и b их содержимое, эквивалентная цепочке присваиваний: $c = a : a = b : b = c$, но не задействующая дополнительную (транзитную) переменную c . Оператор $\text{Swap}(a, b)$ был бы полезен и в среде Mathcad. Но там он нарушил бы установившуюся систему функций и операторов, т. к. он не *возвращает* конкретного значения, а только выполняет некую *процедуру*, цепочку действий. С другой стороны, операторы и функции Mathcad тоже могут ничего не возвращать (например, равенство $a = \sin(x)$), а просто находиться в Mathcad-документе в виде своеобразного комментария (см. разд. 1.3) в ожидании какого-либо действия над ним — решения относительно x , например. Кстати, функция \sin в операторе $a = \sin(x)$ тоже не возвращает значения. Она станет возвращать значение синуса x , если, например, за оператором присваивания последует оператор $a =$. Блок операторов внутри Mathcad-программы также не возвращает значения, а проводит некую цепочку действий, например локальных присваиваний.

Будем считать, что "оператор" и "функция" программно-зависимы, и рассмотрим не их суть, а отличия одного от другого в среде Mathcad. Если же к терминам "опера-

тор" и "функция" все же подходить с позиций математики, которую мы используем в расчетной практике, а не с позиций конкретных особенностей Mathcad, то можно выделить перечисленные далее моменты, довольно условно разделяющие математические инструменты Mathcad на функции и операторы.

- Конкретную функцию из других функций выделяет ее *имя*: $\sin(x)$, $\cos(x)$, $\log(x)$ и т. д. Операторы же отличаются символами: $n!$, $-x$, $|x|$, \int и т. д. Есть в среде Mathcad операторы с невидимым (отсутствующим) символом: x^y (возведение в степень), x_n (элемент массива) или текстовый индекс и $2K$ (умножение). О невидимых операторах Mathcad будет сказано в *разд. 1.2.3*.
- Некоторые операторы Mathcad имеют двойственное содержание. Приведу пример: x^2 — что это? Оператор с двумя операндами возведения в степень, где второй операнд — двойка, или однооперандный оператор возведения в квадрат? Кнопки этих двух операторов можно видеть на рис. 1.1. Второй пример: $2K$ — это оператор умножения или просто переменная с именем $2K$ (а такая переменная возможна в Mathcad, см. *разд. 1.2.2*)? Другие примеры: оператор M^T — это возведение в степень или транспонирование матрицы? Оператор $|M|$ — это определитель матрицы или абсолютное значение (модуль) и т. д.? Есть математические действия, которые реализованы в среде Mathcad и в виде операторов, и в виде функций. Один из примеров — экспонента, которую можно вызывать как e^x (одно- или двухоперандный оператор с невидимым символом, см. выше) и как $\exp(x)$ (функция с аргументом x).
- В среде Mathcad все функции имеют одинаковый статус. Некоторые же операторы выстроены по иерархии. Так, составной оператор $2+2 \cdot 2$ возвращает 6, а не 8, т. к. оператор умножения по приоритету выше оператора сложения. Эта иерархия (возведение в степень, умножение/деление, сложение/вычитание) может как устанавливаться, так и нарушаться скобками.
- Атрибутом (неотъемлемым признаком) функции являются скобки, которые обрамляют список аргументов функции: $\sin(x)$, $\min(1, -7, 5, 4)$, $\text{Find}(a, b)$ и т. д. Скобки сами по себе являются своеобразным оператором как в Mathcad, так и в других вычислительных системах и в математике в целом, меняющим порядок выполнения операторов: $(2 + 2) \cdot 2 = 8$, а не 6, как было бы без скобок (см. выше). Функции с одним или двумя аргументами можно вызвать без скобок — операторами, и об этом будет рассказано далее.

Скобки в операторе могут обрамлять операнд, но не как атрибут, а как новый оператор, объединяющий свои операнды в один операнд: $(5)!$ (факториал пяти — скобки здесь лишние, но к ошибке не приводят), $(1 + 4)!$ (факториал суммы двух чисел — без скобок будет ошибка, вернее, другой, неверный результат, что еще опаснее просто ошибки с соответствующим сообщением). В среде Mathcad вложение одних пар скобок в другие может приводить к автоматической замене круглых скобок на квадратные, что облегчает ориентацию в них, но иногда затрудняет ввод такого оператора с листа, т. к. не всем понятно, что нужно вводить круглые скобки, а не квадратные, которые стоят на месте.

- У оператора число операндов всегда фиксированное. Число операндов меняется от единицы до некоторого числа. Если говорить о традиционных математических операторах, то максимальное их количество равно четырем: определенный интеграл (см. рис. 1.33), сумма и др. Но если считать, что график в Mathcad-документе (см. разд. 1.5) выводится также специальным оператором, то о максимальном числе операндов можно поспорить. В случае с графиком можно также говорить о переменном числе операндов в операторе, часть из которых можно не вводить, значения операторов могут устанавливаться по умолчанию, а потом произвольно меняться.

Некоторые встроенные функции Mathcad имеют переменное число аргументов. Так, функция `root` (она возвращает нуль другой, анализируемой функции — см. главу 3) может иметь либо два, либо четыре аргумента. Фактически это две встроенные функции, скрывающиеся под одним именем и работающие по разным алгоритмам. У функции `log` обычно бывает один аргумент, но если через запятую приписать второй, то он будет менять основание логарифма с 10 (умолчание) на иное, указанное пользователем через заданное значение второго аргумента. В документации к Mathcad (руководство пользователя, help и др.) необязательные аргументы отмечаются квадратными скобками, например `log(z, [b])` — см. главу 3 и рис. 1.41). Функция `Find`, возвращающая решение системы аналитических уравнений и неравенств, может иметь от 1 до 50 аргументов. При этом 50 — это ограничение не конкретной функции `Find`, а всех Mathcad-функций с переменным числом аргументов (например, `stack` и др.).

Есть в среде Mathcad функция, ставшая документированной только в 12-й версии и имеющая число аргументов, равное нулю. Это функция `time`, возвращающая время в секундах, прошедшее с некоторой даты, которая сама по себе не важна, т. к. пользователи работают не с абсолютным значением времени, а с разницей значений, соответствующих двум вызовам этой функции, используя эту разницу для тестирования расчетов, сравнения скоростных характеристик тех или иных групп операторов. О функции `time` говорят, что она имеет *формальный аргумент*, значение которого не влияет на значение, возвращаемое данной функцией. *Формальными* в несколько ином смысле называют аргументы пользовательских функций: $y(x) := \sin(x)$ или $y(t) := \sin(t)$ — здесь может быть любая другая "формальная" переменная, а не только x или t , хотя в документации к Mathcad формальные аргументы функций прописаны особо: n — целое число, x — вещественное число, z — вещественное или комплексное число, v — вектор, M — матрица, A — квадратная матрица и т. д. Это не влияет на ход расчета, но позволяет лучше понимать ту или иную запись в Mathcad-документе.

Математические константы π и e , встроенные в Mathcad, также можно рассматривать как функции с нулевым числом аргументов и возвращающие постоянные величины.

Среди функций и операторов Mathcad есть особые, понять работу которых можно, не только уяснив их математическую суть, но и вникнув в способы их численной реализации в среде Mathcad. Эти функции и операторы возвращают свое значение, зависящее не только (а очень часто и не *столько*) от значений аргументов или операндов,

но и от того, что *находится вблизи* этих функций и операторов, а также от их дополнительной *настройки*. Так, функция Find при неизменных значениях своих аргументов может возвращать совершенно разные значения, зависящие от первого приближения при численном решении систем уравнений. Вторым примером. На рис. 1.33 показано, что появляется в контекстном меню, выпадающем при нажатии правой кнопки мыши на операторе определенного интеграла. Там даны настройки алгоритма для численного решения этой задачи. Можно сказать, что внутрь, в недра некоторых функций проникает что-то еще, минуя "официальный вход" — список аргументов функции. Такие "боковые входы" нередко создает и сам пользователь, формируя, например, новую функцию следующим образом: $a:=2 \quad y(x) := x^2 - a$ вместо более правильной записи $y(x, a) := x^2 - a$. Первая запись имеет то преимущество, что при большом числе аргументов можно выделить один или два из них и записать в списке формальных аргументов создаваемой функции. Остальные можно условно считать константами. Недостаток такой записи заключается в том, что такую незамкнутую ("некапсулированную") функцию трудно переносить в другой документ — можно забыть про некоторые внешние константы, "растерять их по дороге". Такая функция в чем-то подобна Mathcad-документу с некоторыми данными во внешних файлах (см. рис. 1.28). Внешний файл, как мы уже отметили, часто является причиной очень распространенной ошибки: передается Mathcad-файл, а данные к нему передать забывают.

Функции вводятся в документ, как правило, нажатием соответствующих клавиш: например, для функции синус — <s>, <i>, <n>, <> и т. д. Но лучше для этого использовать диалоговое окно **Вставка функции** (рис. 1.34).

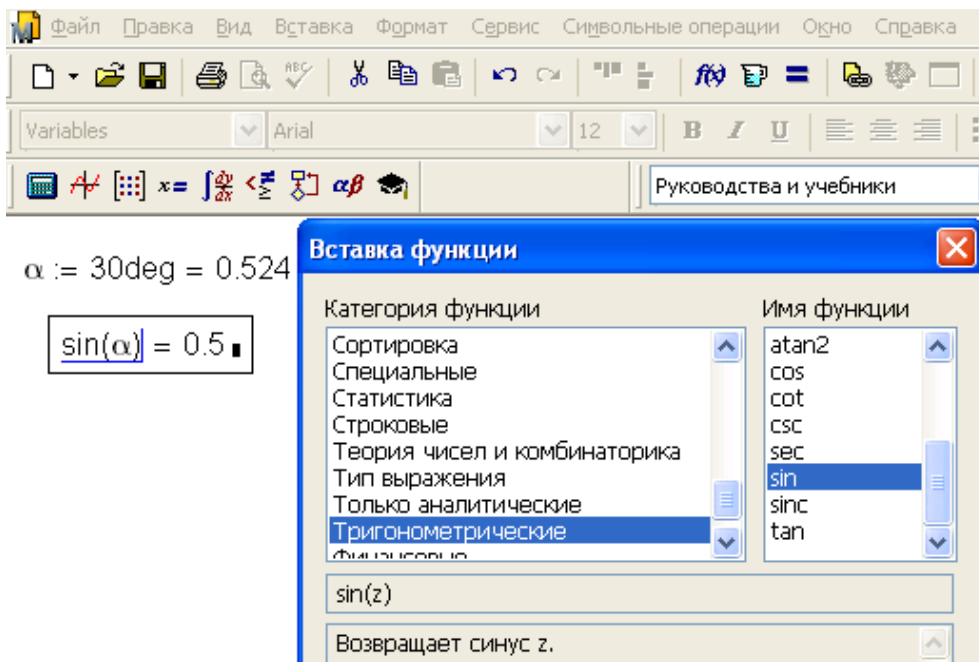


Рис. 1.34. Мастер функций Mathcad

Пользовательские функции в среде Mathcad могут появиться в диалоговом окне **Вставка функции** (функции в категориях **WaterSteamPro** на рис. 1.30) только после того, как их переведут в разряд пользовательских по технологии DLL (см. главу 7).

Встроенные операторы вводятся в Mathcad-документ нажатием кнопок с их изображением (рис. 1.35).

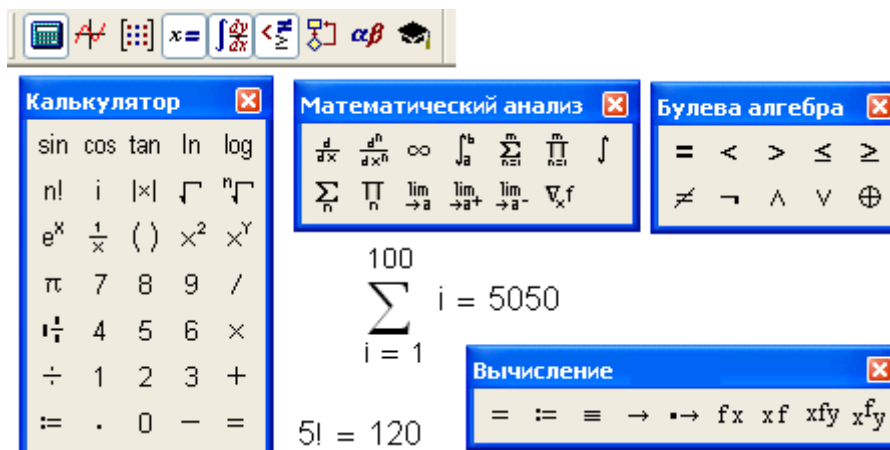


Рис. 1.35. Панели встроенных операторов

Своеобразными операторами, как отмечено в *разд. 1.1*, являются также и операторы интерфейса ввода/вывода.

К инструментам решения задач относят и некоторые *команды меню*. Так, с помощью команды **Символьные операции | Переменная | Решить** можно найти решение уравнения или неравенства, предварительно написав его в рабочем документе и отнеав курсором переменную — неизвестную уравнения или неравенства. Ответ появится ниже (по умолчанию), правее или вместо исходного выражения (после дополнительной настройки). Но сейчас команды меню редко используются для решения задач математики, т. к. их почти полностью заменили операторы. Команды меню оставлены для совместимости версий Mathcad.

Функции (встроенные и пользовательские) с одним или двумя аргументами помимо традиционного способа (набор имени, скобка и т. д.) можно вызывать в Mathcad-документе в виде операторов нажатием кнопок **fx**, **xf** (один аргумент), **xfy** и **xfy** (два аргумента) на панели инструментов **Вычисление** (правый верхний угол рис. 1.35). При этом появляются заготовки *префиксного (fx)* и *постфиксного (xf)* операторов с одним операндом, *инфиксного (xfy)* и *древовидного (xfy)* операторов с двумя операндами. Это позволяет вводить в Mathcad-документы *пользовательские операторы* в том написании, которое человек уже давно использует и к которому он привык. На рис. 1.36—1.41 показаны примеры использования операторов **fx**, **xf**, **xfy** и **xfy** для решения некоторых частных задач.

Функция `mean` в среде Mathcad возвращает среднее арифметическое значение элементов массива (вектора, матрицы) или списка. Первый вызов этой функции на рис. 1.36 сделан в традиционном виде — в виде функции. Из-за этого получились двойные скобки (одна (внутренняя) пара скобок обрамляет элементы вектора, а вторая — аргумент функции), которые могут смущать неопытного пользователя. Он попытается убрать лишние скобки, не понимая, почему этого нельзя сделать. Кроме того, лишняя информация мешает читать документ. Выход из положения — вызов "матричной" функции (функции, у которой аргумент — массив) с одним аргументом не в виде функции, а в виде *префиксного оператора fx* (**f** — символ оператора, а **x** — его операнд), операнд которого не требует лишнего "скобочного" обрамления (второй оператор на рис. 1.36). Так, кстати, можно записать в Mathcad-документе $\sin x$, а не $\sin(x)$ ¹.

¹ Я в самом начале книги отметил, что записи математических выражений в среде Mathcad полностью совпадают аналогичными записями в учебниках и справочниках. Но есть исключения. Так, например, запись $\sin^2 x$ в среде Mathcad будет иметь вид $\sin(x)^2$. С другой стороны, записи в учебниках и справочниках, тоже меняются под давлением компьютеров.

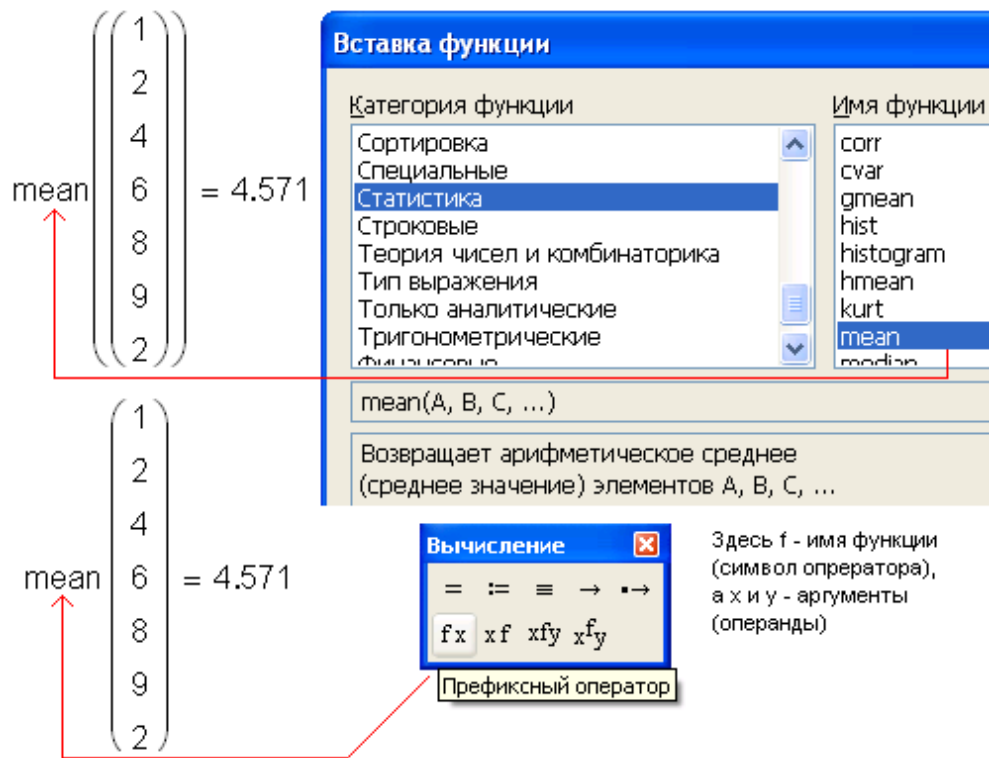


Рис. 1.36. Префиксный оператор

На рис. 1.37 переопределяется встроенный оператор факториала так, чтобы он мог работать и с дробными операндами². Для этого в расчет вводится функция $!(x)$, равная гамма-функции, но с увеличенным на единицу аргументом. Эту новую функцию можно вызывать в традиционном виде $!(5.5)$, но лучше так, как показано на рис. 1.37 — в виде постфиксного оператора ($x!$ — операнд, а затем символ оператора).

Примечание

Возможность введения в Mathcad-документ символа $!$ и других спецсимволов не в виде оператора (факториала в данном случае), а в виде имени функции описана в разд. 1.2.2.

² Это, по правде говоря, не совсем корректное с точки зрения математики действие.

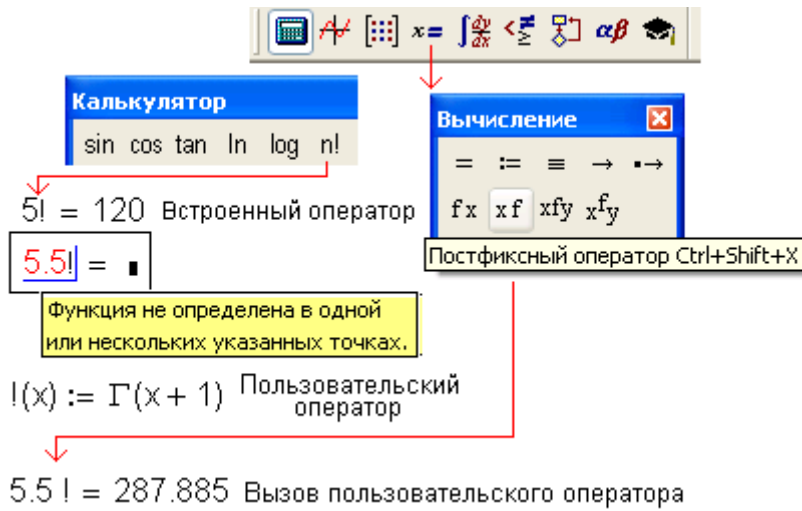


Рис. 1.37. Постфиксный оператор

А вот другой пример использования постфиксного оператора (рис. 1.38), где показано, как имя функции может стать индексом и более удобно отмечать минимальные и максимальные элементы массивов — не $\min(V)$ и $\max(V)$, а V_{\min} и V_{\max} . Здесь переопределенные функции \min и \max имеют в своих именах невидимый символ (см. разд. 1.2.3), а сами являются нижними индексами.

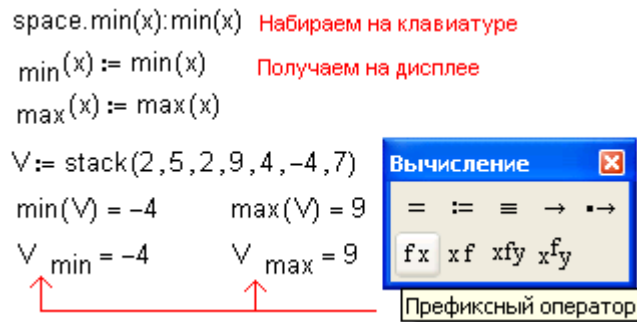


Рис. 1.38. Постфиксный оператор — индекс массива

Примеры, показанные на рис. 1.36—1.38, — простые и не имеют особой практической ценности. А вот использование префиксного и постфиксного операторов для работы с относительными шкалами измерений различных величин очень полезно. Об этом будет рассказано в разд. 1.4.

На рис. 1.39 показано, как с помощью *инфиксного оператора* в расчет вводится дополнительный (см. панель инструментов **Булева алгебра**) двухоператорный булев оператор "приблизительно равно", очень полезный при организации разного рода

итераций в алгоритмах численной математики, где циклы, как правило, прерываются оператором "примерно равно", а не встроенным оператором "точно равно". На рис. 1.39 показано, что π примерно равно 3.142. Значение же 3.14 (а его мы, как правило, и помним) "не равно примерно" π .

Примечание

Булевы функции Mathcad возвращают либо 0 (нет, false), либо 1 (да, true).

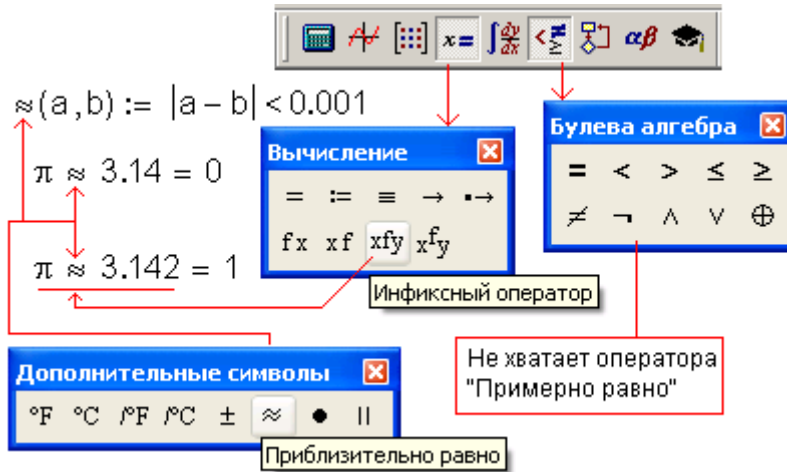


Рис. 1.39. Инфиксный оператор

На рис. 1.40 иллюстрируется работа *древовидного оператора*: сначала переопределяются встроенные двухоперандные операторы сложения и умножения, а затем с помощью древовидного оператора графикой раскрывается иерархия выражения $2 + 2 \cdot 2$, о котором было сказано ранее.

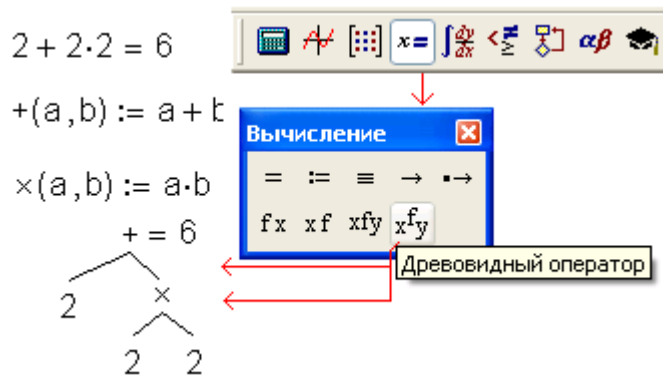


Рис. 1.40. Древовидный оператор

Древовидный оператор в практических расчетах, как правило, не используется, но он очень уместен в документах образовательной направленности, где важен не сам результат, а раскрытие методики его получения через графику — ветви дерева.

Встроенную функцию `log` из-за того, что у нее один или два аргумента, можно вызывать всеми *пятью способами вызова функций* (рис. 1.41): традиционным, через префиксный, постфиксный, инфиксный и древовидный оператор.

Вставка функции

Категория функции	Имя функции
Все	Infit
Аппроксимация и сглаживание кривой	InGamma
Бесселя	LoadColormap
Вейвлет-преобразование	loess
Векторы и матрицы	log
Гиперболические	logfit
Доступ к файлу	logpts
Интерполяция и прогнозирование	logspace
Комплексные числа	lookin

`log(z, [b])`

Возвращает логарифм z по основанию b. Если параметр b не указан, возвращает логарифм по основанию 10.

<div style="border: 1px solid gray; padding: 2px; font-size: small;"> В... = := ≡ → •→ fx xf xfy xfy </div>	<p><code>n := 0.132</code></p> <p><code>log(n) = -0.879</code> Вызов функции</p> <p><code>log n = -0.879</code> Вызов префиксного оператора</p> <p><code>n log = -0.879</code> Вызов постфиксного оператора</p> <p><code>base := 10</code> Основание логарифма</p> <p><code>log(n, base) = -0.879</code> Вызов функции</p> <p><code>n log base = -0.879</code> Вызов инфиксного оператора</p> <p><code>log</code> = -0.879 Вызов древовидного оператора</p> <div style="margin-left: 20px;"> $\begin{array}{c} \text{log} \\ \swarrow \quad \searrow \\ n \quad \text{base} \end{array}$ </div>
--	---

Рис. 1.41. Пять способов вызова функции `log`

Одна из причин популярности Mathcad заключается в том, что пользователь вправе вставлять в документы либо функцию, либо оператор в зависимости от того, к чему он привык, изучая математику в школе или вузе. Благодаря этому Mathcad-документ, как уже не раз отмечалось, максимально похож на лист с математическими выкладками, написанными от руки или созданными в среде какого-либо текстового процессора (Scientific Word, ChiWriter, MathType и др.). Но формулы, введенные с помощью формульных процессоров, не работают, не считаются. Из-за этого в них очень часто вкрадываются ошибки (опечатки).

С другой стороны, наш стиль написания математических выражений "на бумаге" тоже меняется (или должен меняться) под воздействием стиля математических пакетов — Mathcad в первую очередь. Так в настоящее время квадрат синуса все чаще стали записывать не как $\sin^2 x$, а как $\sin(x)^2$ или даже (лучше) как $(\sin x)^2$, не полагаясь на принцип умолчания.

Мы привыкли писать у переменной верхний и нижний индексы прямо друг над другом. В Mathcad (и в Word непосредственно в документе, а не через редактор формул) этого сделать нельзя — эти индексы приходится писать с некоторым сдвигом по горизонтали (см., например, рис. 1.44 с записью формулы химического вещества). Эту особенность стали также учитывать при записи "на бумаге".