# Некоторые фундаментальные типы данных в системе Maple

На протяжении всего курса нам часто придётся сталкиваться с объектами, которые будут называться уравнениями. В Maple уравнения являются встроенным типом данных, не требующим никакого дополнительного объявления.

```
> eq:=A=B;
```

$$eq := A = B$$

Внешне уравнения выглядят как равенства, поэтому ряд функций Maple при попытке узнать тип выражения в качестве ответа выдают результат *равенство*, т.е. `` `=` ``.

```
> whattype(eq);
```

`=`

Это синоним для типа уравнение

```
> type(eq,equation);
```

*true*

Над уравнениями можно выполнять многие привычные действия, которые нам уже известны. Например, операция `` `+` `` позволяет добавить к обеим частям уравнения одну и ту же величину.

```
> eq:=eq+1;
```

$$eq := A + 1 = B + 1$$

Если при выполнении операции `` `+` `` оба слагаемых являются уравнениями, то происходит почленное сложение левых и правых частей уравнения. Вот как это выглядит

```
> EQ:=A-1=B-1;
  eq+EQ;
```

$$EQ := A - 1 = B - 1$$

$$2\,A = 2\,B$$

Совершенно ясно, что уравнения можно не только складывать, но и вычитать.

```
> eq-EQ;
```

$$2 = 2$$

Как поменять знаки обеих частей уравнения на противоположные? Оказывается, достаточно умножить наше уравнение на -1.

```
> eq*(-1);
```

$$-A - 1 = -B - 1$$

Если оба сомножителя в произведении являются уравнениями, то возникает сообщение об ошибке. Почленное произведение частей уравнения выполняется другим способом.

```
> eq*EQ;
Error, (in simpl/reloprod) invalid terms in product: (A+1 = B+1)*(A-1 = B-1)
```

Это же замечание касается и операции деления.

```
> eq/EQ;
Error, (in simpl/reloprod) invalid terms in product: (A-1 = B-1)^-1
```

**Однако данные ошибки возникают только в относительно старых версиях Maple!** Современные версии системы компьютерной алгебры позволяют перемножать уравнения, а также делить соответствующие части уравнений друг на друга. Если Вам часто приходится сталкиваться с такими преобразованиями, то, возможно, стоит подумать о переходе на более новую версию.

Уравнения всегда можно делить на скалярную величину.

```
> eq:=eq/(A+1);
```

$$eq := 1 = \frac{B + 1}{A + 1}$$

Для доступа к частям уравнения имеются две специальные функции: *lhs* (left-hand side) и *rhs* (right-hand side). С их помощью можно извлечь нужную часть и выполнить над ней любую операцию, которая допустима для скалярных выражений.

Например, так можно получить новое уравнение из уже имеющегося путём возведения в квадрат обеих его частей.

```
> EQ2:=lhs(EQ)^2=rhs(EQ)^2;
```

$$EQ2 := (A-1)^2 = (B-1)^2$$

Обсудим ещё один фундаментальный тип данных, который идейно близок к уравнениям. Этот тип данных называется диапазон и синтаксически обозначается с помощью двух подряд идущих точек `..`. Слева от этого двоеточия записывается нижняя граница диапазона, а справа – верхняя. Ниже приводится простой пример диапазона.

```
> R1:=2..5;
```

$$R1 := 2..5$$

Диапазоны используются при указании границ в операциях суммирования и интегрирования, а также в ряде циклических конструкций. В частности, если требуется последовательность из 10 первых натуральных чисел, то её можно создать следующей конструкцией:

```
> $1..10;
```

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10$$

Диапазон не обязан задаваться явно. Вместо него можно использовать переменную.

```
> $R1;
```

$$2, 3, 4, 5$$

Вот более сложный пример генерации последовательности квадратов натуральных чисел.

```
> k^2$k=R1;
```

$$4, 9, 16, 25$$

С точки зрения внутреннего представления объекты типа `..` и range идентичны. Действительно,

```
> whattype(R1);
```

$$..$$

в то же время

```
> type(R1,range);
```

$$true$$

Арифметические операции над объектами типа range не допускаются. Впрочем, это тоже касается **только относительно старых версий**.

```
> evalr(R1+1);
Error, invalid terms in sum: 2 .. 5
```

Однако с помощью функций *lhs* и *rhs* можно извлечь границы диапазона и выполнить над ними необходимые преобразования.

```
> R2:=lhs(R1)+1..rhs(R1)+2;
```

$$R2 := 3..7$$

Границы диапазона не обязаны быть целыми числами.

```
> $2.4..9.8;
```

$$2.4, 3.4, 4.4, 5.4, 6.4, 7.4, 8.4, 9.4$$

В цикле можно перечислять и символы

```
> $"a".."h";
```

"a", "b", "c", "d", "e", "f", "g", "h"

Как правило, нижняя граница диапазона не превышает верхнюю. Однако в некоторых циклических конструкциях допускается использование шага, с которым изменяется параметр цикла. В таких ситуациях нижняя граница будет превышать верхнюю.

```
> seq(10..5,-1);
```

$$10, 9, 8, 7, 6, 5$$

С помощью диапазонов легко сгенерировать большой набор переменных.

```
> 'x||k'$k=1..6;
```

$$x1, x2, x3, x4, x5, x6$$

Если мы хотим не просто создать эти переменные, но и присвоить им нулевые значения, то следует записать следующий оператор присваивания

```
> cat(x,2..6):=0$5;
```

$$x2, x3, x4, x5, x6 := 0, 0, 0, 0, 0$$

```
> x2;x3;x4;x5;
```

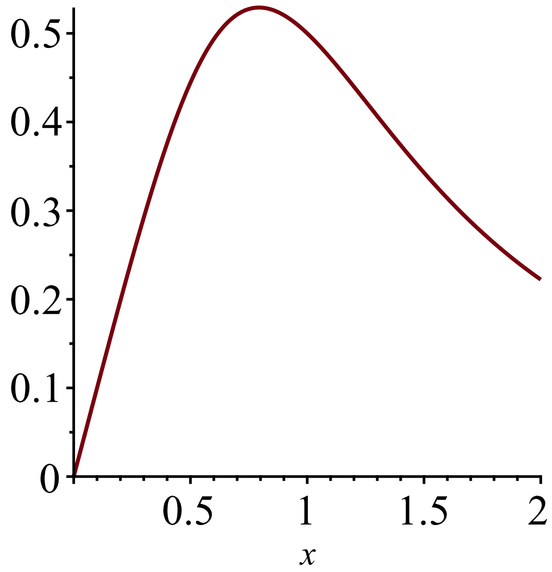
$$0$$

$$0$$

$$0$$
$$0$$

Во вводных курсах диапазоны часто используются при построении графиков. С их помощью можно указать в каких пределах будет изменяться аргумент функции.

```
> plot(x/(1+x^3),x=0..2);
```




## Основные команды, применяемые для упрощения выражений

При решении задач часто приходится сталкиваться с выражениями, которые содержат дроби или рациональные функции (или же комбинацию тех и других, связанных знаками арифметических операций). Упрощение таких выражений следует начинать с применения команды *normal.* Эта команда приводит исходное выражение к нормальной форме, которая имеет вид *числитель/знаменатель*. В процессе приведения выражения к такому виду система попытается представить *числитель и знаменатель* как два взаимно простых многочлена с целыми коэффициентами. При этом будет применяться небольшой набор преобразований с рациональными функциями. Преобразования позволяют упрощать суммы, произведения, а также целочисленные степени констант и переменных.

С точки зрения пользователя применение этой команды выглядит как приведение исходного выражения к общему знаменателю и упрощение числителя результирующей дроби. Благодаря тому, что набор преобразований небольшой, команда *normal* работает значительно быстрее многих других команд упрощения.

```
> normal(1/x+x/(x+1));
```

$$\frac{x^2+x+1}{x\,(x+1)}$$

Её можно применять и в тех случаях, когда упрощаемое выражение не содержит знаменателя.

```
> normal(x^2-(x+1)*(x-1)-1);
```

$$0$$

Если аргументом *normal* является уравнение, диапазон или другой аналогичный объект, то команда рекурсивно применяется ко всем компонентам этого объекта.

```
> normal(1/x-1/x^2=5/(x+1)-4/(x-1));
```

$$\frac{x-1}{x^2}=\frac{x-9}{(x+1)\,(x-1)}$$

Поскольку в результате применения команды *normal* числитель и знаменатель должны быть взаимно просты, то при обнаружении в ходе преобразований общих множителей произойдёт их сокращение.

```
> normal((x^2-y^2)/(x-y)^3);
```

$$\frac{x+y}{(x-y)^2}$$

Когда исходное выражение содержит радикалы, степени с нецелым показателем или какие-либо другие функции, то *normal* рекурсивно применяется к их аргументам, а затем полученный результат "замораживается" с целью получения нормальной формы.

```
> expr:=sqrt(2)*(x-a)/(2*x-a)-((sqrt(x)/(sqrt(2*x)+sqrt(a)))^2+
        ((sqrt(2*x)+sqrt(a))/(2*sqrt(a)))^(-1))^(1/2);
```

$$expr := \frac{\sqrt{2}\,(x-a)}{2x-a} - \sqrt{\frac{x}{\left(\sqrt{2}\,\sqrt{x}+\sqrt{a}\right)^2} + \frac{2\sqrt{a}}{\sqrt{2}\,\sqrt{x}+\sqrt{a}}}$$

Следует, однако, отметить, что без дополнительных предположений существенных упрощений добиться невозможно. И дело тут не в возможностях самой команды *normal*,

```
> normal(expr);
```

$$-\frac{\sqrt{\frac{2\sqrt{2}\,\sqrt{x}\,\sqrt{a}+2a+x}{\left(\sqrt{2}\,\sqrt{x}+\sqrt{a}\right)^2}}\;a - 2\sqrt{\frac{2\sqrt{2}\,\sqrt{x}\,\sqrt{a}+2a+x}{\left(\sqrt{2}\,\sqrt{x}+\sqrt{a}\right)^2}}\;x - \sqrt{2}\,a + \sqrt{2}\,x}{-2x+a}$$

а в том, что необходимо указывать **область допустимых значений** исходного выражения!

```
> normal(expr) assuming positive;
```

$$-\frac{1}{(-2x+a)\left(\sqrt{2}\,\sqrt{x}+\sqrt{a}\right)}\Big(-2\sqrt{x}\,a+2x^{3/2}-a^{3/2}\sqrt{2}+\sqrt{a}\,\sqrt{2}\,x$$
$$+\sqrt{2\sqrt{2}\,\sqrt{x}\,\sqrt{a}+2a+x}\;a-2\sqrt{2\sqrt{2}\,\sqrt{x}\,\sqrt{a}+2a+x}\;x\Big)$$

По умолчанию, как показано в предыдущем примере, *normal* раскрывает скобки в числителе,

```
> Expr:=(x+sqrt(3))/(sqrt(x)+sqrt(x+sqrt(3)))+
        (x-sqrt(3))/(sqrt(x)-sqrt(x-sqrt(3)));
```

$$Expr := \frac{x+\sqrt{3}}{\sqrt{x}+\sqrt{x+\sqrt{3}}} + \frac{x-\sqrt{3}}{\sqrt{x}-\sqrt{x-\sqrt{3}}}$$

оставляя знаменатель в факторизованном виде.

```
> normal(Expr);
```

$$\frac{\sqrt{3}\,\sqrt{x+\sqrt{3}}+\sqrt{3}\,\sqrt{x-\sqrt{3}}-x\sqrt{x+\sqrt{3}}+x\sqrt{x-\sqrt{3}}-2x^{3/2}}{\left(\sqrt{x}+\sqrt{x+\sqrt{3}}\right)\left(-\sqrt{x}+\sqrt{x-\sqrt{3}}\right)}$$

Если требуется раскрыть скобки и в знаменателе, то в команде *normal* следует указать опцию expanded.

```
> normal(Expr,expanded);
```

$$\frac{\sqrt{3}\,\sqrt{x+\sqrt{3}}+\sqrt{3}\,\sqrt{x-\sqrt{3}}-x\sqrt{x+\sqrt{3}}+x\sqrt{x-\sqrt{3}}-2x^{3/2}}{-x+\sqrt{x}\,\sqrt{x-\sqrt{3}}-\sqrt{x+\sqrt{3}}\,\sqrt{x}+\sqrt{x+\sqrt{3}}\,\sqrt{x-\sqrt{3}}}$$

При работе с дробями часто возникает необходимость выделить из неё числитель или же знаменатель. Для обращения к числителю в системе Maple предусмотрена функция *numer.* Знаменатель дроби может быть найден с помощью функции *denom.*

В простейших случаях результаты выполнения этих команд очевидны.

```
> v:=2/x;
  numer(v);denom(v);
```

$$v := \frac{2}{x}$$

$$2$$

$$x$$

Когда аргументом является целое число или же число с плавающей точкой, то знаменатель равен 1.

```
> denom(2);denom(0.5);
```

$$1$$

$$1.0$$

Если же аргументом является комплексное число, у которого как вещественная, так и мнимая части рациональны, то эти части приводятся к общему знаменателю.

```
> v:=1/2+I/3;w:=1/2.0+I/3.0;
  numer(v);numer(w);
  denom(v);denom(w);
```

$$v := \frac{1}{2} + \frac{1}{3}\,\mathrm{I}$$

$$w := 0.5000000000 + 0.3333333333\,\mathrm{I}$$

$$3 + 2\,\mathrm{I}$$

$$0.5000000000 + 0.3333333333\,\mathrm{I}$$

$$6$$

$$1.0$$

Выражения, имеющие тип отличный от числового (*numeric*), **как правило**, предварительно нормализуются. В качестве результата возвращаются числитель и знаменатель нормальной формы.

```
> v:=((p^(1/4)-q^(1/4))^(-2)+(p^(1/4)+q^(1/4))^(-2))/
     ((sqrt(p)+sqrt(q))/(p-q));
```

$$v := \frac{\left(\frac{1}{\left(p^{1/4} - q^{1/4}\right)^2} + \frac{1}{\left(p^{1/4} + q^{1/4}\right)^2}\right)(p-q)}{\sqrt{p} + \sqrt{q}}$$

```
> numer(v);denom(v);
```

$$2\left(\sqrt{p} + \sqrt{q}\right)(p-q)$$

$$\left(p^{1/4} - q^{1/4}\right)^2 \left(p^{1/4} + q^{1/4}\right)^2 \left(\sqrt{p} + \sqrt{q}\right)$$

Выполняемая командами *numer* и *denom* нормализация является облегчённой версией команды *normal*. В связи с этим из результатов иногда не удаляются общие множители.

```
> w:=1/x^3-(x^2-x+1)/x^3;
  numer(w);denom(w);
```

$$w := \frac{1}{x^3} - \frac{x^2 - x + 1}{x^3}$$

$$-x\,(x-1)$$

$$x^3$$

В то же время более тщательная проверка даёт другой результат

```
> w:=normal(w);
  numer(w);denom(w);
```

$$w := -\frac{x-1}{x^2}$$

$$-x + 1$$

$$x^2$$

Для нормализации выражений, содержащих радикалы, в системе Maple предусмотрена команда *radnormal*. Она способна выполнять преобразования над радикалами, записанными как в явной форме,

```
> expr:=root[3]((10-7*sqrt(2))/(10+7*sqrt(2)));
```

$$expr := \frac{\left(\left(10 - 7\sqrt{2}\right)\left(10 + 7\sqrt{2}\right)^2\right)^{1/3}}{10 + 7\sqrt{2}}$$

```
> radnormal(expr);
```

$$\frac{\sqrt{2} + 2}{10 + 7\sqrt{2}}$$

так и в неявной, то есть через функцию *RootOf*.

```
> w:=RootOf(_Z^2+_Z+1,index=1)-(-1)^(2/3);
```

$$w := RootOf\left(\_Z^2 + \_Z + 1, index = 1\right) - (-1)^{2/3}$$

```
> radnormal(w);
```

$$0$$

Если аргументом *radnormal* оказывается список, множество, уравнение и другие подобные им объекты, то команда рекурсивно применяется ко всем компонентам этих объектов.

```
> R2:=root[3](5*sqrt(2)-7)..root[4](28+16*sqrt(3));
```

$$R2 := \left(5\sqrt{2}-7\right)^{1/3}..\left(28+16\sqrt{3}\right)^{1/4}$$

```
> radnormal(R2);
```

$$\sqrt{2}-1..1+\sqrt{3}$$

В некоторых случаях команда *radnormal* позволяет уменьшить вложенность радикалов, однако подобное упрощение не гарантируется.

```
> Expr:=root[3](9+sqrt(80))+root[3](9-sqrt(80));
```

$$Expr := \left(9+4\sqrt{5}\right)^{1/3}+\left(9-4\sqrt{5}\right)^{1/3}$$

```
> radnormal(Expr);
```

$$3$$

Если аргументом *radnormal* является константное выражение, представленное в виде дроби, то знаменатель этой дроби **не всегда** рационализируется.

```
> expr:=(4/(3-sqrt(5)))^2-((6-5*sqrt(6))/(5-sqrt(6)))^2;
```

$$expr := \frac{16}{\left(3-\sqrt{5}\right)^2}-\frac{\left(6-5\sqrt{6}\right)^2}{\left(5-\sqrt{6}\right)^2}$$

```
> radnormal(expr);
```

$$-\frac{2\left(9\sqrt{5}-17\right)}{3\sqrt{5}-7}$$

Можно заставить *radnormal* избавиться от радикалов в знаменателе, если указать опцию *'rationalized'*.

```
> radnormal(expr,rationalized);
```

$$8+6\sqrt{5}$$

Для полиномов, представленных в факторизованном виде, *radnormal* пытается сохранить частичную факторизацию. Скобки внутри каждого множителя раскрываются и коэффициенты нормализуются.

```
> Expr:=(x*(x+1)-6^(1/2))*(x^2-2^(1/2)*3^(1/2)+x);
```

$$Expr := \left(x\,(x+1)-\sqrt{6}\right)\left(x^2-\sqrt{2}\sqrt{3}+x\right)$$

```
> radnormal(Expr);
```

$$\left(x^2-\sqrt{2}\sqrt{3}+x\right)^2$$

Если мы хотим раскрыть все скобки, то как и при использовании команды *normal* следует указать опцию expanded.

```
> radnormal(Expr,expanded);
```

$$6-2\sqrt{3}\sqrt{2}\,x^2+x^4-2\sqrt{3}\sqrt{2}\,x+2x^3+x^2$$

```
> v:=((p^(1/4)-q^(1/4))^(-2)+(p^(1/4)+q^(1/4))^(-2))/
     ((sqrt(p)+sqrt(q))/(p-q));
```

$$v := \frac{\left(\frac{1}{\left(p^{1/4}-q^{1/4}\right)^2}+\frac{1}{\left(p^{1/4}+q^{1/4}\right)^2}\right)(p-q)}{\sqrt{p}+\sqrt{q}}$$

Нормализация рациональных функций, содержащих радикалы, выполняется по тем же правилам, как и при вызове команды *normal*.

```
> radnormal(v);
```

$$\frac{2\,(p-q)}{p-2\sqrt{p}\sqrt{q}+q}$$

```
> Ex:=((1+sqrt(x))/sqrt(1+x)-sqrt(1+x)/(1+sqrt(x)))^2-((1-sqrt(x))
  /sqrt(1+x)-sqrt(1+x)/(1-sqrt(x)))^2;
```

$$Ex := \left(\frac{1+\sqrt{x}}{\sqrt{x+1}} - \frac{\sqrt{x+1}}{1+\sqrt{x}}\right)^2 - \left(\frac{1-\sqrt{x}}{\sqrt{x+1}} - \frac{\sqrt{x+1}}{1-\sqrt{x}}\right)^2$$

```
> radnormal(Ex);
```

$$-\frac{16\,x^{3/2}}{\left(x^2-2\,x+1\right)\,(x+1)}$$

Опция expanded раскрывает скобки в знаменателе.

```
> radnormal(Ex,expanded);
```

$$-\frac{16\,x^{3/2}}{x^3-x^2-x+1}$$

Можно узнать, что делал (или что пытался сделать) *radnormal*, если установить значение переменной *infolevel* равным положительной константе больше 1.

```
> infolevel[radnormal]:=5;
```

$$infolevel_{radnormal} := 5$$

```
> radnormal(Ex);
radnormal: entering radnormal at time 1.263
radnormal: The RootOfs may not be independent
radnormal: field is {RootOf(_Z^2-x index = 1)}
radnormal: extension degree is 2
radnormal: substitutions are [radnormal/Radical(x 2 0) = RootOf(_Z^2-x
index = 1)]
radnormal: backward substitutions are [RootOf(_Z^2-x index = 1) = x^
(1/2)]
radnormal: exiting radnormal at time  1.263
```

$$-\frac{16\,x^{3/2}}{\left(x^2-2\,x+1\right)\,(x+1)}$$

В ранних версиях Maple для упрощения выражений с радикалами рекомендовалось использование команды *radsimp*.

Однако при работе этой функции использовались преобразования, которые **не гарантируют** корректность полученного результата на всей комплексной плоскости. В связи с этим в относительно недавних версиях системы Maple данная функция относится к разряду *нерекомендуемых*.

Вот типичный пример работы *radsimp*

```
> radsimp((x^2-2*x+1)^(1/2));
```

$$x-1$$

Этот результат неверен, если $x < 1$.

Для сохранения работоспособности ранее написанных скриптов в нынешних версиях системы происходит прозрачная для пользователя подмена вызова *radsimp* на вызов процедуры *simplify* с опциями radical и symbolic. Когда все неизвестные принимают неотрицательные значения, результат оказывается таким же, как и обращении к *radnormal*.

```
> v:=((p^(1/4)-q^(1/4))^(-2)+(p^(1/4)+q^(1/4))^(-2))/
     ((sqrt(p)+sqrt(q))/(p-q));
```

$$v := \frac{\left(\frac{1}{\left(p^{1/4}-q^{1/4}\right)^2} + \frac{1}{\left(p^{1/4}+q^{1/4}\right)^2}\right)(p-q)}{\sqrt{p}+\sqrt{q}}$$

```
> radsimp(v);
```

$$\frac{2\,(p-q)}{\left(p^{1/4}-q^{1/4}\right)^2\left(p^{1/4}+q^{1/4}\right)^2}$$

При упрощении дробей и рациональных функций *radsimp*, как и команда *radnormal*, не пытается рационализировать знаменатели.

```
> radsimp(1/(1+sqrt(2)));
```

$$\frac{1}{1+\sqrt{2}}$$

Для принудительного избавления от радикалов в знаменателе следует указывать опцию ratdenom.

```
> radsimp(1/(1+sqrt(2)),ratdenom);
```

$$\sqrt{2}-1$$

```
> Expr:=sqrt((p^2-q*sqrt(p))/(sqrt(p)-root[3](q))+p*root[3](q))*(p+
  root[6](p^3*q^2))^(-1/2);
```

$$Expr := \frac{\sqrt{\frac{p^2-q\sqrt{p}}{\sqrt{p}-q^{1/3}}+p\,q^{1/3}}}{\sqrt{p+\left(p^3\,q^2\right)^{1/6}}}$$

Если рациональная функция не содержит вложенных радикалов, то результат приводится к нормальной форме. При наличии вложенности алгоритм применяется рекурсивно.

```
> radsimp(Expr);
```

$$\frac{\left(q^{1/3}+\sqrt{p}\right)p^{1/4}}{\sqrt{p+\left(p^3\,q^2\right)^{1/6}}}$$