Глава 3. Програмирование в маткад.

Для вставки программного кода в документы в Mathcad имеется специальная панель инструментов **Programming** (Программирование), которую можно вызвать на экран нажатием кнопки **Programming Toolbar** на панели **Math** (Математика), как показано на рис. 1. Большинство кнопок этой панели выполнено в виде текстового представления операторов программирования, поэтому их смысл легко понятен.

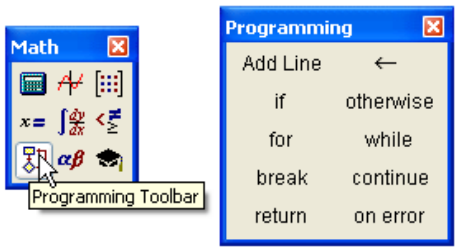

Рис. 1. Панель инструментов Programming

Основными инструментами работы в Mathcad являются математические выражения, переменные и функции. Нередко записать формулу, использующую ту или иную внутреннюю логику (например возвращение различных значений в зависимости от условий), в одну строку не удается. Назначение программных модулей как раз и заключается в определении выражений, переменных и функций в несколько строк, часто с применением специфических программных операторов.

Листинг 3. Функция условия, определенная с помощью программы:

$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{"positive"} & \text{if } x > 0 \\ \text{"zero"} & \text{otherwise} \end{cases}$$

$$f(1) = \text{"positive"}$$

$$f(-1) = \text{"negative"}$$

$$f(0) = \text{"zero"}$$

Программирование имеет ряд существенных преимуществ, которые в ряде случаев делают документ более простым и читаемым:

- возможность применения циклов и условных операторов;
- простота создания функций и переменных, требующих нескольких простых шагов (как в примере листинга 6.3);
- возможность создания функций, содержащих закрытый для остального документа код, включая преимущества использования локальных переменных и обработку исключительных ситуаций (ошибок).

Как видно из листинга 3, программный модуль обозначается в Mathcad вертикальной чертой, справа от которой последовательно записываются операторы языка программирования.

Создание программы

Чтобы создать программный модуль (**Add Line**), например, представленный в предыдущем разделе (см. листинг 3):

- Введите часть выражения, которая будет находиться слева от знака присваивания и сам знак присваивания. В нашем примере это имя функции f (x).

- При необходимости вызовите на экран панель инструментов **Programming** (Программирование) (см. рис. 1).
- Нажмите на этой панели кнопку **Add Line** (Добавить линию).
- Если приблизительно известно, сколько строк кода будет содержать программа, можно создать нужное количество линий повторным нажатием кнопки **Add Line** (Добавить линию) соответствующее число раз (на рис. 2 показан результат трехкратного нажатия).
- В появившиеся местозаполнители введите желаемый программный код, используя программные операторы. В рассматриваемом примере в каждый местозаполнитель вводится строка, например, "positive11 (рис. 3), затем нажимается кнопка **If** (Если) на панели **Programming** (Программирование) и в возникший местозаполнитель вводится выражение x>0 (рис. 4).

После того как программный модуль полностью определен и ни один местозаполнитель не остался пустым, функция может использоваться обычным образом, как в численных, так и в символьных расчетах.

Не вводите с клавиатуры имена программных операторов. Для их вставки можно применять лишь сочетания клавиш, которые приведены в тексте всплывающей подсказки (рис. 2 и 3).

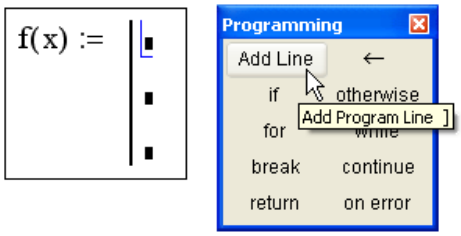

Рис. 2. Начало создания программного модуля

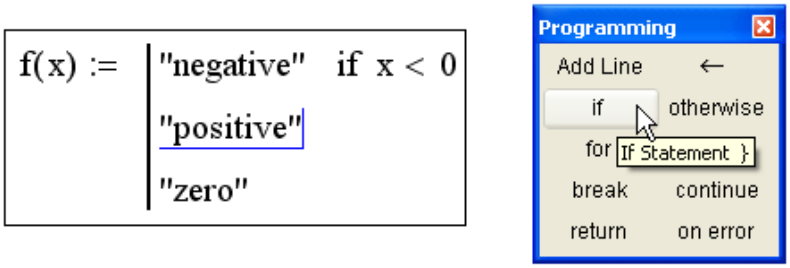

Рис. 3. Вставка программного оператора

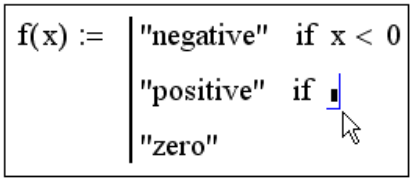

Рис. 4. Вставка условия в программу

Разработка программы

Вставить строку программного кода в уже созданную программу можно в любой момент с помощью той же самой кнопки **Add Line** (Добавить линию). Для этого следует предварительно поместить на нужное место внутри программного модуля линии ввода. Например, расположение линии ввода на строке, показанной на рис. 5, приведет к появлению новой линии с местозаполнителем перед этой строкой. Если передвинуть вертикальную линию ввода из начала строки (как на рис. 5) в ее конец, то новая линия появится после строки. Если выделить строку не целиком, а лишь некоторую ее часть (рис.

6), то это повлияет на положение в программе новой строки кода (результат нажатия кнопки **Add Line** показан на рис. 7).

Не забывайте, что для желаемого размещения линий ввода внутри формулы можно использовать не только мышь и клавиши со стрелками, но и пробел. С помощью последовательных нажатий пробела линии ввода "захватывают" разные части формулы.

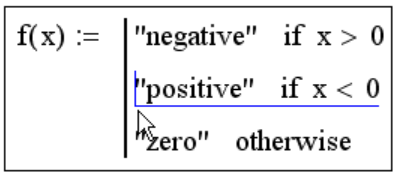

Рис. 5. Вставка новой строки в существующую программу

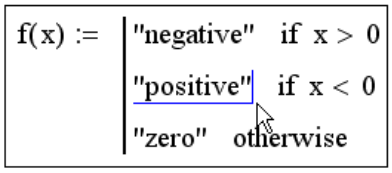

Рис. 6. Положение линий ввода влияет на положение новой линии

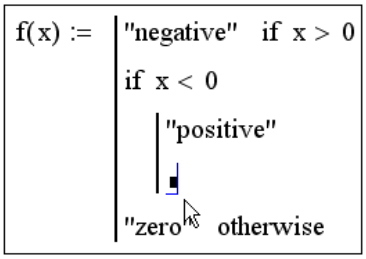

Рис. 7. Результат вставки новой линии в программу (из положения рис. 6)

Зачем может потребоваться вставка новой линии в положение, показанное на рис. 7. Новая вертикальная черта с двумя линиями выделяет фрагмент программы, который относится к условию x>0, находящемуся в его заголовке. Пример возможного дальнейшего программирования показан в листинге 4.

Листинг 4. Пример усовершенствования программы:

```
f(x) := "negative"  if x < 0
        if x > 0
            "positive"
            "big positive"  if x > 1000
        "zero"  otherwise

f(1) = "positive"
f(10^5) = "big positive"
```

В режиме выполнения программы, а это происходит при любой попытке вычислить f (x), выполняется последовательно каждая строка кода. Например, в предпоследней строке листинга 4 вычисляется f(i). Рассмотрим работу каждой строки кода этого листинга.

- Поскольку x=1, то условие x<0 не выполнено, и в первой строке ничего не происходит.
- Условие второй строки x>0 выполнено, поэтому выполняются обе следующие строки, объединенные короткой вертикальной чертой в общий фрагмент.
- Функции f(x) присваивается значение f(x) = "positive".

- Условие x>1000 не выполнено, поэтому значение "big positive" не присваивается f (x), она так и остается равной строке "positive".
- Последняя строка не выполняется, т. к. одно из условий (x>0) оказалось истинным, и оператор **otherwise** (т. е. "иначе") не понадобился.

Таким образом, основной принцип создания программных модулей заключается в правильном расположении строк кода. Ориентироваться в их действии довольно легко, т. к. фрагменты кода одного уровня сгруппированы в программе с помощью вертикальных черт.

Локальное присваивание. Условные операторы.

Язык программирования Mathcad не был бы эффективным, если бы не позволял создавать внутри программных модулей локальные переменные, которые "не видны" извне, из других частей документа. Присваивание в пределах программ, в отличие от документов Mathcad, производится с помощью оператора **Local Definition** (Локальное присваивание), который вставляется нажатием кнопки с изображением стрелки ← на панели **Programming** (Программирование).

Ни оператор присваивания **:=**, ни оператор вывода **=** в пределах программ не применяются.

Локальное присваивание иллюстрируется листингом 5. Переменная z существует только внутри программы, выделенной вертикальной чертой. Из других мест документа получить ее значение невозможно.

Листинг 5. Локальное присваивание в программе:

$$f(x) := \left| \begin{array}{l} z \leftarrow 4 \\ z + x \end{array} \right.$$

$$f(1) = 5$$

Условные операторы (if, otherwise)

Действие условного оператора **if** состоит из двух частей. Сначала проверяется логическое выражение (условие) справа от него. Если оно истинно, выполняется выражение слева от оператора **if**. Если ложно – ничего не происходит, а выполнение программы продолжается переходом к ее следующей строке. Вставить условный оператор в программу можно следующим образом (см. рис. 8):

- Если необходимо, введите левую часть выражения и оператор присваивания.
- Создайте новую строку программного кода, нажав на панели **Programming** (Программирование) кнопку **Add Line** (Добавить строку).
- Нажмите кнопку условного оператора **if**.
- Справа от оператора **if** введите условие. Пользуйтесь логическими операторами, вводя их с панели **Boolean** (Булевы операторы).
- Выражение, которое должно выполняться, если условие истинно, введите слева от оператора **if**.
- Если в программе предусматриваются дополнительные условия, добавьте в программу еще одну строку нажатием кнопки **Add Line** и введите их таким же образом, используя оператор **if** или **otherwise**.

Оператор **otherwise** используется совместно с одним или несколькими условными операторами **if** и указывает на выражение, которое будет выполняться, если ни одно из условий не оказалось истинным. Примеры использования операторов **if** и **otherwise** приведены в предыдущих разделах (см. листинги 3 и 4).

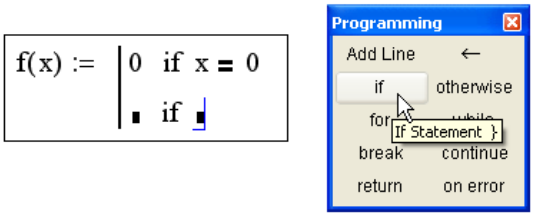

Рис. 8. Вставка условного оператора

Операторы цикла

Рассмотрим операторы цикла – **for**, **while**, **break**, **continue**.

В языке программирования Mathcad имеются два оператора цикла: **for** и **while**. Первый из них дает возможность организовать цикл по некоторой переменной, заставляя ее пробегать некоторый диапазон значений. Второй создает цикл с выходом из него по некоторому логическому условию.

Чтобы вставить в программный модуль оператор цикла:

- Создайте в программном модуле новую линию.
- Вставьте один из операторов цикла **for** или **while** нажатием одноименной кнопки на панели **Programming** (Программирование).
- Если выбран оператор **for** (рис. 9), то вставьте в соответствующие местозаполнители имя переменной и диапазон ее значений (листинги 6 и 7), а если **while** – то логическое выражение, при нарушении которого должен осуществляться выход из цикла (листинг 8).

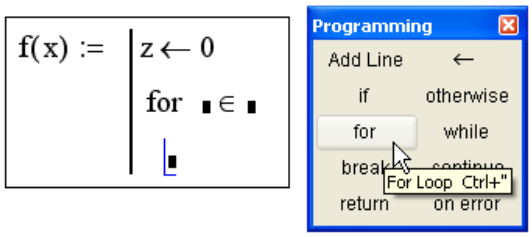

Рис. 9. Вставка оператора цикла

- В нижний местозаполнитель введите тело цикла, т. е. выражения, которые должны выполняться циклически.

При необходимости дополните программу другими строками и введите в них нужный код.

Диапазон значений переменной в условии цикла **for** можно задать как с помощью диапазона ранжированной переменной (листинг 6), так и с помощью вектора (листинг 7).

Листинг 6. Оператор цикла for с ранжированной переменной:

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ \quad z \leftarrow z + i \end{array} \right.$$

$$x = 15$$

Листиг 7. Оператор цикла for с вектором:

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \text{for } i \in (1 \;\; 2 \;\; 3) \\ \quad z \leftarrow z + i \end{array} \right.$$

$$x = 6$$

Листинг 8. Опратор цикла while:

```
x := | z ← 0
     | while z < 10
     |   z ← z + 1

x = 10
```

Возврат значения

Если для определения переменной или функции применяется программный модуль, то его строки исполняются последовательно при вычислении в документе этой переменной или функции. Соответственно, по мере выполнения программы рассчитываемый результат претерпевает изменения. В качестве окончательного результата выдается последнее присвоенное значение (примеры можно найти в листингах 3-12). Чтобы подчеркнуть возврат программным модулем определенного значения, можно взять за правило делать это в последней строке программного модуля (листинг 13).

Листинг 13. Возврат значения обозначен явно в последней строке программы:

```
f(x) := | y ← x^2
        | z ← y + 1
        | z

f(2) = 5
```

Вместе с тем, можно прервать выполнение программы в любой ее точке (например с помощью условного оператора) и выдать некоторое значение, применив оператор **return**. В этом случае при выполнении указанного условия (листинг 14) значение, введенное в местозаполнитель после **return**, возвращается в качестве результата, а никакой другой код больше не выполняется. Вставляется в программу оператор **return** с помощью одноименной кнопки панели **Programming** (Программирование).

Листинг 14. Применение оператора return:

```
f(x) := | z ← x^2
        | return "zero"  if x = 0
        | return "i"  if x = i
        | z

f(−1) = 1

f(2) = 4

f(0) = "zero"

f(i) = "i"
```

Перехват ошибок

Программирование в Mathcad позволяет осуществлять дополнительную обработку ошибок (**on error**). Если пользователь предполагает, что выполнение кода в каком-либо месте программного модуля способно вызвать ошибку (например деление на ноль), то эту ошибку можно перехватить с помощью оператора **on error**. Чтобы вставить его в программу, надо поместить линии ввода в ней в нужное положение и нажать кнопку с

именем оператора **on error** на панели **Programming** (Программирование). В результате появится строка с двумя местозаполнителями и оператором **on error** посередине (рис. 10).

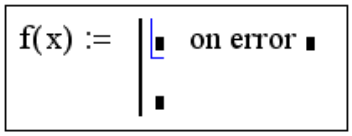

Рис. 10. Вставка оператора перехода по ошибке

В правом местозаполнителе следует ввести выражение, которое должно выполняться в данной строке программы. В левом – выражение, которое будет выполнено вместо правого выражения, если при выполнении последнего возникнет ошибка. Приведем пример применения оператора on error (листинг 15) в программном модуле, который рассчитывает функцию обратного числа значению n. Если n=0, то и присвоенное значение z=0, поэтому в последней строке программы выполняется правое выражение расчета i/z. Так происходит при расчете f (-2). Если попытаться вычислить f (0) как в конце листинга, то выполнение программы, заложенной в f (n), вызовет ошибку деления на ноль в последней строке программы. Соответственно, вместо выражения справа от оператора **on error** будет выполнено левое выражение, присваивающее функции f(n) строковое значение "user error: cannot divide by zero" (пользовательская ошибка: деление на ноль невозможно). Конечно, этой строке можно присвоить и текст на русском языке.

Листинг 15. Перехват ошибки деления на ноль:

$$f(n) := \begin{| z \leftarrow n \\ \text{"user error: can't divide by zero"} \quad \text{on error } \frac{1}{z} \end{|}$$

$$f(-2) \rightarrow \frac{-1}{2}$$

$$f(0) = \text{"user error: can't divide by zero"}$$

Оператор перехвата ошибок удобно применять в комбинации со встроенной функцией **error(S)**. Она приводит к генерации ошибки в обычной для Mathcad форме с сообщением s. Пример усовершенствования листинга 15 для такого стиля обработки ошибки деления на ноль показан на рис. 11.

$$f(n) := \begin{| z \leftarrow n \\ \text{error("user error: can't divide by zero")} \quad \text{on error } \frac{1}{z} \end{|}$$

$$f(-2) \rightarrow \frac{-1}{2}$$

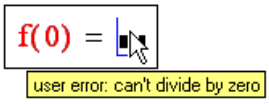

Рис. 11. Перехват ошибки деления на ноль

Обратите внимание, что сделанные изменения свелись к помещению текста сообщения об ошибке в аргумент функции **error**.

Рассмотрим два простых примера использования программных модулей в Mathcad для численных (листинг 16) и символьных (листинг 17) расчетов. В двух приведенных листингах используется большинство операторов, рассмотренных в данной главе. Когда вы станете сами разрабатывать свои программные модули в Mathcad, не забывайте, что

операторы программирования вставляются в текст программы с помощью кнопок панели инструментов **Programming** (Программирование).

Их имена нельзя ни в коем случае просто набивать на клавиатуре, поскольку они не будут восприняты Mathcad корректно.

Листинг 16. Программирование в численных расчетах:

$$f(n) := \left| \begin{array}{l} \text{return } -99 \text{ if } n < 0 \\ z \leftarrow 1 \\ \text{for } i \in 1..n \\ \quad z \leftarrow z \cdot i \\ z \end{array} \right.$$

$f(-2) \rightarrow -99$

$f(0) = 0$

$f(3.9) = 6$

$f(3) = 6$

$f(10) = 3.629 \times 10^6$

Листинг 17. Программирование в символьных расчетах:

$$f(n) := \left| \begin{array}{l} -1 \text{ if } n < 0 \\ x \text{ on error } \dfrac{d^n}{dx^n} x^{10} \text{ otherwise} \end{array} \right.$$

$f(1) \rightarrow 10 \cdot x^9$

$f(10) \rightarrow 3628800$

$f(-3) \rightarrow -1$

$f(2.1) \rightarrow x$