

А. М. Абухба,

Абхазский государственный университет, г. Сухум, Абхазия

НЕКОТОРЫЕ ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ В MATHCAD PRIME

Аннотация

В статье рассмотрены некоторые отличительные особенности программирования в системе компьютерной математики Mathcad Prime. Приведены примеры различных программ, способствующих пониманию сути этих особенностей.

Ключевые слова: Mathcad, программирование, подпрограмма, переменная, операторы, функция, вектор, матрица, массив, ветвление, циклы.

DOI: 10.32517/2221-1993-2018-17-9-28-33

Контактная информация

Абухба Андрей Мустафаевич, доцент кафедры прикладной математики и информатики Абхазского государственного университета, г. Сухум, Абхазия; *адрес:* 384900, Республика Абхазия, г. Сухум, ул. Университетская, д. 1; *телефон:* +7 (840) 223-10-19; *e-mail:* aabuchba@mail.ru

A. M. Abukhba,
Abkhaz State University, Sukhum, Abkhazia

SOME FEATURES OF PROGRAMMING IN MATHCAD PRIME

Abstract

Some distinctive features of programming in the system of computer mathematics Mathcad Prime are considered in the article. Examples of various programs that contribute to understanding the essence of these features are given.

Keywords: Mathcad, programming, subroutine, variables, operators, functions, vector, matrix, array, branching, cycles.

Система Mathcad — одна из самых мощных и эффективных систем компьютерной математики, которая занимает особое место среди подобных других систем (Matlab, Maple, Mathematica и др.).

Mathcad — это приложение для математических и инженерных вычислений, промышленный стандарт проведения, распространения и хранения расчетов [5]. Задумывался Mathcad как средство «программирования без программирования», но, если возникает такая потребность, Mathcad имеет довольно простые для усвоения инструменты программирования, позволяющие строить весьма сложные алгоритмы, к чему прибегают, когда встроенных средств решения задачи не хватает, а также когда необходимо выполнять серийные расчеты [6]. Начиная с версии Mathcad Prime 3.0 появилась возможность внешнего программирования: теперь можно создавать программы на C++ или FORTRAN за пределами Mathcad (например, в Microsoft Visual Studio). Внутри Mathcad эти программы будут работать как пользовательские функции.

Основное отличие Mathcad от аналогичных программ — это графический, а не текстовый режим ввода выражений. В любом случае формулы будут иметь привычный, аналогичный книжному, вид. Любые переменные, формулы, параметры можно изменять, наблюдая воочию соответствующие изменения результата [6]. Также имеется возможность использования различных единиц измерения при проведении расчетов. Mathcad предоставляет большой набор единиц измерения, соответствующих международной системе единиц измерения СИ, американскому аналогу USCS и традиционной системе типа СГС (сантиметр — грамм — секунда).

Совместное использование функций и операторов Mathcad позволяет составлять довольно сложные программы. Если учесть, что в Mathcad содержится более 700 встроенных функций, то становится очевидным, насколько упрощается процесс составления программ для решения сложных задач. При составлении программ можно организовать вызов встроенных констант и функций, а также обращение к определенным ранее переменным и функциям пользователя. В этом смысле программирование на встроенном в систему языке программирования можно назвать **укрупненным программированием**.

На простых примерах, не отвлекаясь на решение сложных задач, рассмотрим некоторые особенности использования встроенного языка программирования.

Для составления программ (подпрограмм-функций) в Mathcad Prime имеется специальная вкладка **Программирование** (рис. 1):

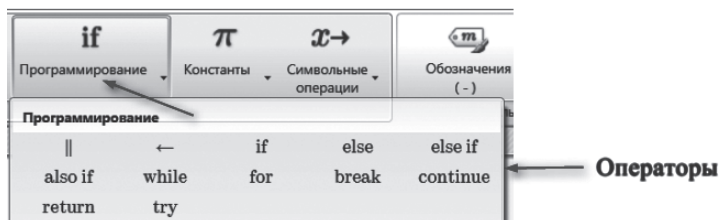


Рис. 1

Алгоритмические конструкции (операторы) в среде Mathcad вводятся не традиционным набором через клавиатуру служебных слов if, then, else, while и т. д., а нажатием одной из соответствующих кнопок панели инструментов или с использованием сочетания клавиш, предусмотренных для этих целей.

Необходимо отметить, что использование в программах обычного оператора присваивания ($:=$) вызовет ошибку. Вместо него необходимо использовать оператор локального присваивания $\leftarrow$. Также в программах

Программа	Результаты работы программы
$Kvur(a, b, c) := \left\ \begin{array}{l} D \leftarrow b^2 - 4 \cdot a \cdot c \\ x_1 \leftarrow \frac{-b + \sqrt{D}}{2 \cdot a} \\ x_2 \leftarrow \frac{-b - \sqrt{D}}{2 \cdot a} \\ \text{if } D < 0 \\ \quad \ Res \leftarrow \text{“Комплексные корни”} \\ \text{else} \\ \quad \ Res \leftarrow \text{“Действительные корни”} \\ \left[\begin{array}{l} Res \\ x_1 \\ x_2 \end{array} \right] \end{array} \right\ $	$Kvur(1, 2, 3) = \left[\begin{array}{l} \text{“Комплексные корни”} \\ \left[\begin{array}{l} -1 + 1.414i \\ -1 - 1.414i \end{array} \right] \end{array} \right]$ $Kvur(1, 2, -3) = \left[\begin{array}{l} \text{“Действительные корни”} \\ \left[\begin{array}{l} 1 \\ -3 \end{array} \right] \end{array} \right]$

Рис. 2

в определениях условия (логическое выражение) нельзя использовать обычный знак равенства $=$, вместо него необходимо использовать оператор сравнения $\equiv$ (жирный знак равенства).

Создание программ осуществляется с помощью оператора «программа — $\|$ ». Этот оператор является многоступенчатым контейнером для управляющих операторов Mathcad.

Общий вид программы:

$$Name(Param) := \left\| \begin{array}{l} S \\ Res \end{array} \right\|$$

Здесь:

Name — имя программы;

Param — формальные параметры (могут отсутствовать);

S — оператор(ы);

Res — результат.

Mathcad-программа возвращает результат на *последнем шаге*. Причем в качестве результата может возвращаться *единственный «объект»*. «Объект» может представлять собой:

- любое число, в том числе комплексное;
- текстовое сообщение, заключенное в кавычки (“ ”);
- вектор или матрицу (допускается вложенность).

Аналогами одномерного массива являются вектор-строка и вектор-столбец, а двумерного массива — матрица.

Пример 1.*

Найти корни квадратного уравнения:

$$ax^2 + bx + c = 0, a \neq 0.$$

Решение представлено на рисунке 2. В программе на этом рисунке:

Kvur — имя программы;

a, b, c — формальные параметры;

(1, 2, 3) и (1, 2, -3) — фактические параметры, подставляемые вместо формальных.

Вывод полученных результатов записан в виде вектора-столбца, элементами которого являются текстовое сообщение и вектор-столбец, содержащий корни уравнения x_1 и x_2 (вложенность массивов).

Для организации разветвлений в программах на Mathcad Prime используются операторы if, else, else if и also if. Первым оператором должен быть if, который является «заголовком» начала разветвляющегося процесса. Организовать ветвление можно и без использования вкладки *Программирование*. Для этого используется управляющий оператор if, общий вид которого: if (*B*, *S1*, *S2*), что соответствует конструкции: if *B* then *S1* else *S2* (если *B*, то *S1*, иначе *S2*), где *B* — логическое выражение (условие), *S1* и *S2* — произвольные числа, массивы или строки либо функции, при вычислении которых получается результат одного из этих типов. Управляющий оператор if необходимо набирать с помощью клавиатуры.

Пример 2.

Найти значение функции:

$$y = \begin{cases} \sqrt{x}, & \text{при } x \geq 3; \\ x^2, & \text{при } x < 3. \end{cases}$$

Решение.

$$\begin{aligned} y(x) &:= \text{if}(x \geq 3, \sqrt{x}, x^2) \\ y(9) &= 3 \\ y(-5) &= 25 \end{aligned}$$

Рис. 3

Пример 3.

Рекурсивное вычисление факториала $n!$:

$$\begin{aligned} Fakt(n) &:= \text{if}(n = 0, 1, n \cdot Fakt(n - 1)) \\ Fakt(0) &= 1 \\ Fakt(10) &= 3628800 \end{aligned}$$

Рис. 4

* Этот и все рассмотренные в дальнейшем примеры решены в системе Mathcad Prime 4.0.

Программа	Результат работы программы
<pre> ORIGIN := 1 F(n) := $\left\ \begin{array}{l} k \leftarrow 1 \\ \text{for } i \in 1..n \\ \quad \left\ \begin{array}{l} \text{for } j \in 1..n \\ \quad \left\ \begin{array}{l} M1_{i,j} \leftarrow \text{trunc}(\text{rnd}(10)) \\ M2_{i,j} \leftarrow \text{trunc}(\text{rnd}(20)) \\ M3_{i,j} \leftarrow \text{trunc}(\text{rnd}(15)) \end{array} \right. \\ \quad \text{for } i \in [M1 \ M2 \ M3] \\ \quad \quad \left\ \begin{array}{l} MAX_k \leftarrow \max(i) \\ \Delta_k \leftarrow i \\ SrAr_k \leftarrow \text{mean}(i) \\ k \leftarrow k+1 \end{array} \right. \\ \quad \left[\begin{array}{l} M1 \\ M2 \\ M3 \end{array} \right] \quad MAX \ \Delta \ SrAr \end{array} \right\$ </pre>	$F(5) = \left[\begin{array}{c} \left[\begin{array}{ccccc} 0 & 9 & 2 & 2 & 0 \\ 2 & 3 & 1 & 6 & 1 \\ 5 & 7 & 1 & 4 & 9 \\ 5 & 2 & 3 & 3 & 2 \\ 1 & 6 & 3 & 7 & 0 \end{array} \right] \\ \left[\begin{array}{ccccc} 12 & 5 & 7 & 18 & 3 \\ 0 & 0 & 17 & 18 & 11 \\ 5 & 1 & 12 & 7 & 7 \\ 12 & 17 & 18 & 14 & 13 \\ 4 & 1 & 2 & 13 & 0 \\ 1 & 14 & 3 & 13 & 11 \\ 1 & 2 & 2 & 13 & 14 \\ 5 & 8 & 9 & 8 & 2 \\ 0 & 10 & 5 & 11 & 5 \\ 9 & 0 & 1 & 9 & 6 \end{array} \right] \\ \left[\begin{array}{ccc} 9 & \begin{bmatrix} -2302 \\ 15511 \\ 39966 \end{bmatrix} & \begin{bmatrix} 3.36 \\ 8.68 \\ 6.48 \end{bmatrix} \end{array} \right] \end{array} \right]$

Рис. 5

Для организации циклов в программах Mathcad Prime используются операторы цикла с предусловием **while** и с параметром **for**. При этом если количество повторений тела цикла известно или можно вычислить, то используют цикл с параметром, в противном случае — цикл с предусловием.

Цикл с параметром **for** в среде Mathcad намного более гибок, чем его аналоги в языках программирования Pascal, C и т. д. [1].

Общий вид оператора **for**:

$$\left\| \begin{array}{l} \text{for } i \in D \\ S \end{array} \right\|$$

Здесь:

i — любое допустимое имя переменной или параметр цикла;

D — значение или последовательность значений. Чаще всего D — переменная-диапазон, но можно также использовать вектор или список из скаляров или векторов, задающих последовательность скаляров, которые по очереди становятся значениями переменной i . Чтобы выполнить один и тот же расчет над несколькими матрицами, можно определить D как последовательность матриц;

S — тело цикла, любое допустимое выражение или последовательность выражений.

Пример 4.

Создать три квадратные матрицы случайных целых чисел. Для каждой из них найти:

- максимальный элемент;
- определитель;
- среднее арифметическое.

Решение представлено на рисунке 5. В программе на этом рисунке: параметр i в цикле **for** $i \in [M1, M2, M3]$ будет поочередно принимать значения элементов сформированных матриц.

Вывод полученных результатов записан в виде вектора-строки, элементами которого являются вектор-столбец, содержащий исходные матрицы, и векторы-столбцы, содержащие искомые значения для каждого из матриц (вложенность).

Для случая, когда D — переменная-диапазон, в Mathcad Prime предусмотрены две разновидности данного оператора: обычный и модифицированный.

Обычный оператор:

$$\left\| \begin{array}{l} \text{for } i \in N..K \\ S \end{array} \right\|$$

В обычном операторе:

i — параметр цикла;

N, K — начальное и конечное значения параметра цикла (границы диапазона);

S — оператор(ы), или тело цикла.

Параметр цикла i после каждого выполнения S меняется с шагом, равным 1, причем если начальное значение параметра меньше конечного, то увеличивается на 1, в противном случае уменьшается на 1.

Модифицированный оператор:

$$\left\| \begin{array}{l} \text{for } i \in N, N+h..K \\ S \end{array} \right\|$$

В модифицированном операторе h — шаг изменения параметра i : параметр цикла i после каждого выполнения S меняется на величину h .

Необходимо отметить, что параметр цикла может принимать не только целое, но и дробное (десятичные и обыкновенные дроби) значение.

Пример 5.

Вычислить: $S = 1 + 2 + \dots + n$.

Решение.

$$\begin{array}{l}
 \text{Sum}(n) := \left\| \begin{array}{l} S \leftarrow 0 \\ \text{for } i \in 1..n \\ \left\| \begin{array}{l} S \leftarrow S+i \\ S \end{array} \right\| \end{array} \right\| \\
 \text{Sum}(40) = 820
 \end{array}$$

Рис. 6

Пример 6.Вычислить: $S = n + (n - 1) + \dots + 1$.

Решение.

$$\begin{array}{l}
 \text{Sum}(n) := \left\| \begin{array}{l} S \leftarrow 0 \\ \text{for } i \in n..1 \\ \left\| \begin{array}{l} S \leftarrow S+i \\ S \end{array} \right\| \end{array} \right\| \\
 \text{Sum}(40) = 820
 \end{array}$$

Рис. 7

Пример 7.Вычислить: $S = \frac{1}{10} + \frac{2}{10} + \dots + \frac{n}{10}$.

Решение.

$$\begin{array}{l}
 \text{Sum}(n) := \left\| \begin{array}{l} S \leftarrow 0 \\ \text{for } i \in \frac{1}{10}, \frac{2}{10} \dots \frac{n}{10} \\ \left\| \begin{array}{l} S \leftarrow S+i \\ S \end{array} \right\| \end{array} \right\| \\
 \text{Sum}(50) = 127.5
 \end{array}$$

Рис. 8

В примере 5 происходит автоматическое увеличение параметра цикла i на 1, в примере 6 — уменьшение на 1, в примере 7 — увеличение на $\frac{1}{10}$.

Встроенный в Mathcad язык программирования можно использовать и как язык для первоначального ознакомления с азами программирования, так как в нем реализованы все базовые алгоритмические конструкции: следование, ветвление и циклы. Также имеются развитые средства для обработки одномерных и двумерных массивов, предусмотрены возможности создания подпрограмм-функций для их использования (вызова) в других программах и реализации рекурсивных алгоритмов.

В случае использования языка для начального ознакомления или закрепления ранее полученных знаний при изучении других языков программирования желательно составлять развернутые программы, даже тогда, когда имеются встроенные функции и операторы, по-

Программа	Замена	Программа	Замена
$P = x!$, где $x \in N$ $\text{Fakt}(x) := \left\ \begin{array}{l} \text{if } x = 0 \\ \left\ \begin{array}{l} P \leftarrow 1 \end{array} \right\ \\ \text{else} \\ \left\ \begin{array}{l} P \leftarrow 1 \\ \text{for } i \in 1..x \\ \left\ \begin{array}{l} P \leftarrow P \cdot i \end{array} \right\ \end{array} \right\ \\ P \end{array} \right\ $	$x!$	$P = x^y$, где $y \in N$ $\text{Stepen}(x, y) := \left\ \begin{array}{l} \text{if } y = 0 \\ \left\ \begin{array}{l} P \leftarrow 1 \end{array} \right\ \\ \text{else} \\ \left\ \begin{array}{l} P \leftarrow 1 \\ \text{for } i \in 1..y \\ \left\ \begin{array}{l} P \leftarrow P \cdot x \end{array} \right\ \end{array} \right\ \\ P \end{array} \right\ $	x^y
$\text{НОД}(a, b)$ $\text{NOD}(a, b) := \left\ \begin{array}{l} \text{while } a \neq b \\ \left\ \begin{array}{l} \text{if } a > b \\ \left\ \begin{array}{l} a \leftarrow a - b \end{array} \right\ \\ \text{else} \\ \left\ \begin{array}{l} b \leftarrow b - a \end{array} \right\ \end{array} \right\ \\ a \end{array} \right\ $	$\text{gcd}(a, b)$	$\text{НОК}(a, b)$ $\text{NOK}(a, b) := \left\ \begin{array}{l} a1 \leftarrow a \\ b1 \leftarrow b \\ \text{while } a1 \neq b1 \\ \left\ \begin{array}{l} \text{if } a1 > b1 \\ \left\ \begin{array}{l} a1 \leftarrow a1 - b1 \end{array} \right\ \\ \text{else} \\ \left\ \begin{array}{l} b1 \leftarrow b1 - a1 \end{array} \right\ \end{array} \right\ \\ \text{RES} \leftarrow \frac{a \cdot b}{a1} \\ \text{RES} \end{array} \right\ $	$\text{lcm}(a, b)$

Рис. 9

Программа	Замена	Программа	Замена
Наибольшее из двух $Max(a, b) := \begin{cases} \text{if } a > b \\ \quad \quad Max \leftarrow a \\ \text{else} \\ \quad \quad Max \leftarrow b \\ \quad \quad Max \end{cases}$	$\max(a, b)$	Наименьшее из двух $Min(a, b) := \begin{cases} \text{if } a < b \\ \quad \quad Min \leftarrow a \\ \text{else} \\ \quad \quad Min \leftarrow b \\ \quad \quad Min \end{cases}$	$\min(a, b)$
$C_n^k = \frac{n!}{k!(n-k)!}$ $P(m) := \begin{cases} \text{if } m = 0 \\ \quad \quad P \leftarrow 1 \\ \text{else} \\ \quad \quad P \leftarrow 1 \\ \quad \quad \text{for } i \in 1..m \\ \quad \quad \quad P \leftarrow P \cdot i \\ \quad \quad P \end{cases}$ $C(n, k) := \begin{cases} P1 \leftarrow P(n) \\ P2 \leftarrow P(k) \\ P3 \leftarrow P(n-k) \\ \quad \quad Res \leftarrow \frac{P1}{P2 \cdot P3} \\ \quad \quad Res \end{cases}$	$\text{combin}(n, k)$	$A_n^k = \frac{n!}{(n-k)!}$ $P(m) := \begin{cases} \text{if } m = 0 \\ \quad \quad P \leftarrow 1 \\ \text{else} \\ \quad \quad P \leftarrow 1 \\ \quad \quad \text{for } i \in 1..m \\ \quad \quad \quad P \leftarrow P \cdot i \\ \quad \quad P \end{cases}$ $A(n, k) := \begin{cases} P1 \leftarrow P(n) \\ P2 \leftarrow P(n-k) \\ \quad \quad Res \leftarrow \frac{P1}{P2} \\ \quad \quad Res \end{cases}$	$\text{permut}(n, k)$
Максимальный элемент одномерного массива ORIGIN := 1 $Max(n) := \begin{cases} \text{for } i \in 1..n \\ \quad \quad A_i \leftarrow \text{trunc}(\text{rnd}(50)) \\ \quad \quad Max \leftarrow A_1 \\ \quad \quad \text{for } i \in 2..n \\ \quad \quad \quad \text{if } Max < A_i \\ \quad \quad \quad \quad Max \leftarrow A_i \\ \quad \quad [A \quad Max] \end{cases}$	$\max(A)$	Минимальный элемент одномерного массива ORIGIN := 1 $Min(n) := \begin{cases} \text{for } i \in 1..n \\ \quad \quad A_i \leftarrow \text{trunc}(\text{rnd}(50)) \\ \quad \quad Min \leftarrow A_1 \\ \quad \quad \text{for } i \in 2..n \\ \quad \quad \quad \text{if } Min > A_i \\ \quad \quad \quad \quad Min \leftarrow A_i \\ \quad \quad [A \quad Min] \end{cases}$	$\min(A)$

Рис. 10

звolyающие без какого-либо программирования получать необходимые результаты.

Некоторые программы и заменяющие их операторы и функции представлены на рисунках 9, 10.

Практически каждая составленная (автономная) программа, может быть использована при составлении более сложных программ в качестве подпрограммы-функции.

Пример 8.

Вычислить сумму:

$$S = 1! + 2! + \dots + n!, \quad n \geq 1.$$

Решение представлено на рисунке 11.

Переменные, значения которых определены в программе, являются локальными, и их значения недоступны вне этой программы.

Подпрограмма	Основная программа	Результат работы
$P(k) := \begin{array}{ l} P \leftarrow 1 \\ \text{for } i \in 1..k \\ \quad P \leftarrow P \cdot i \\ P \end{array}$	$S(n) := \begin{array}{ l} S \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad S \leftarrow S + P(i) \\ S \end{array}$	$S(5) = 153$

Рис. 11

Пример 9.

$$f := \begin{array}{|l} i \leftarrow 5 \\ i = ? \end{array}$$

Рис. 12

Переменные, значения которых определены вне программы, являются глобальными, и их значения доступны этой программе. При этом переменная должна получить свое значение до его использования в программе с помощью обычного оператора присваивания $\leftarrow$ или после записанной программы с помощью оператора глобального определения $\equiv$. Вышесказанное касается также определения функции пользователя.

Пример 10.

$i := 5$ $Prog := \begin{array}{ l} i \leftarrow i + 1 \\ Prog = 6 \end{array}$	$Prog := \begin{array}{ l} i \leftarrow i + 1 \\ Prog = 6 \\ i \equiv 5 \end{array}$
---	--

Рис. 13

Пример 11.

$f(x) := x^4$ $Prog := \begin{array}{ l} I \leftarrow \int_1^3 f(x) dx \\ Prog = 48.4 \end{array}$	$Prog := \begin{array}{ l} I \leftarrow \int_1^3 f(x) dx \\ Prog = 48.4 \\ f(x) \equiv x^4 \end{array}$
--	---

Рис. 14

* * *

В данной статье рассмотрены те особенности программирования в Mathcad Prime, с которыми изучающий его так или иначе будет сталкиваться на начальном этапе. Понимание сути этих особенностей существенно облегчит процесс начала составления различных программ в этой системе. Хорошими помощниками в этом являются также встроенные в систему русифицированные справочник и учебник.

Mathcad, помимо своего главного предназначения — выполнения разнообразных научных и инженерных расчетов, также прекрасно подходит для организации обучения основам программирования.

Mathcad является эффективным средством в сфере образования, поскольку он прост в использовании, широко применим во многих областях науки и техники.

Список использованных источников

1. Воскобойников Ю. Е., Задорожный А. Ф. Основы вычислений и программирования в пакете Mathcad PRIME: учебное пособие. СПб.: Лань, 2016.
2. Кирьянов Д. В. Mathcad 15/Mathcad Prime 1.0. СПб.: БХВ-Петербург, 2012.
3. Основы вычислений и программирования в пакете Mathcad: учебное пособие / под ред. Ю. Е. Воскобойникова. Новосибирск: НГАСУ (Сибстрин), 2012.
4. Уроки Mathcad // Archicad-autocad.com — Ваш графический софт. <http://archicad-autocad.com/uroki-mathcad.html>
5. Mathcad // ПТС — Официальный дистрибьютер Creo, Windchill, Mathcad, Vericut в России. <http://pts-russia.com/products/Mathcad/>
6. Mathcad // Википедия, Свободная энциклопедия. <https://ru.wikipedia.org/wiki/Mathcad>

НОВОСТИ

Amazon запустил сервис для создания и тестирования софта в области робототехники

Компания Amazon запустила облачный сервис RoboMaker, с помощью которого разработчики в сфере робототехники смогут создавать дополнительный софт, тестировать его и внедрять в свою продукцию. Об этом говорится на сайте компании.

С помощью инструментов RoboMaker разработчики смогут довольно просто создавать приложения для работы своей техники, а также тестировать ее внутри

виртуальной среды Amazon и пользоваться базовой инфраструктурой компании.

Операционная система роботов (ROS), с которой и работает RoboMaker, является наиболее известным софтом для робототехники с открытым кодом. По данным Amazon, RoboMaker позволит не только разрабатывать софт, но и быстро его тестировать благодаря облачным технологиям компании.

(По материалам «Хайтек»)