

Прикладное программное обеспечение

Тема 1.16. Программирование в MathCAD

Учебные вопросы:

- 1. Операторы языка программирования.
- 2. Составление и редактирование текста программы.
- 3. Анимация графиков.

Литература

Основная:

- 1. Макаров Е. Инженерные расчеты в MathCad 14. – СПб.: Питер, 2007.
- 2. Макаров Е. Инженерные расчеты в MathCad 15. – СПб.: Питер, 2011.
- 3. Информатика. Базовый курс: учебное пособие / ред. С. В. Симонович. - 3-е изд. - СПб.: Питер, 2013.

Литература

Дополнительная:

- 1 Общий курс высшей математики для экономистов: учебник : [гриф Мин. обр.] / Б. М. Рудык и др.; ред. В. И. Ермаков. - М. : ИНФРА-М, 2010.
- 2. Практикум по высшей математике для экономистов: учебное пособие: [гриф Мин. обр.] : [гриф УМО] / Н. Ш. Кремер и др.; ред. Н. Ш. Кремер. - М.: ЮНИТИ-ДАНА, 2003.

1. Операторы языка программирования.

Пакет Mathcad содержит собственные скромные по составу, но универсальные средства программирования. Эти средства позволяют запрограммировать практически любой алгоритм, хотя они и не являются столь удобными и мощными по сравнению с обычными системами программирования. Назначение и обозначения операторов языка программирования Mathcad схожи с обычными языками программирования, но структура и правила записи операторов существенно отличаются от Си подобных языков программирования.

К числу основных операторов языка программирования относят следующие.

- 1. Оператор добавления линии (строки) программы **add**. В результате выбора этого оператора в текст программы добавляется вертикальная линия, справа от которой в шаблоне ■ записывается очередной оператор преобразования данных.
- 2. Оператор локального присваивания ←

Пример

$a \leftarrow 1234.$

$$z \leftarrow \int_0^x e^{-x} dx$$

- 3. Условный оператор *<выражение>* **if** *<условие>*

В такой форме «выражение» (выражение и условие в программе записываются без угловых скобок) будет вычисляться только, если условие является истинным, в противном случае оператор соответствует пустому.

Пример

- $1+2*y$ if $x>0$

- 4. Оператор цикла **for**. Шаблон оператора

for ■ € ■.. ■

■

В левый верхний шаблон заносят имя переменной целого типа, в шаблоны за знаком € заносят параметры цикла.

Выражение, помещенное в нижележащий шаблон ■, будет выполняться заданной число раз. Пример записи фрагмента цикла

for $z \in N_{\min} .. N_{\max}$

$x \leftarrow x + z$

- 5. Оператор цикла **while**. Шаблон оператора **while** ■



В верхний шаблон заносится условие. Условие можно задать в виде булева или арифметического выражения, например

```
while i < 5
```

```
while x+3.1
```

В последнем случае цикл будет выполняться, пока $x+3.1$ не равно нулю.

В нижний шаблон помещается один или несколько операторов (путем добавления строк программы), которые следует выполнять, пока условие является истинным.

6. Оператор выбора **otherwise**. Применяется совместно с условным оператором `if`. Позволяет сформировать условный оператор в форме, соответствующей полной записи условного оператора обычного языка программирования. Пример совместного применения операторов `if` и `otherwise` для вычисления абсолютного значения скалярной величины

- `x` if `x > 0`

- `-x` otherwise

7. Оператор прерывания программы или перехода в конец цикла **break**.

8. Оператор продолжения программы (цикла) **continue**. 8. 9.

Оператор выхода из блока программы с передачей результата.

return <выражение>

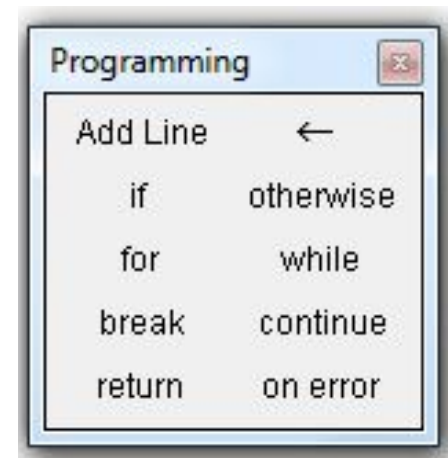
•Пример

```
return x/2
```

2. Составление и редактирование текста программы.

Составление текста программы на языке пакета существенно отличается от записи программ на обычном языке программирования.

1. Операторы программы *нельзя* набирать вручную. Их следует вводить щелчками мыши по наименованиям операторов в палитре программирования **Programming**.
2. Программы являются *функциями*, результат выполнения функции передается посредством необязательного оператора return



3. Переменные, объявленные выше или левее текста программы, передаются в программу *по значению* и не изменяются, даже если по ходу программы им присваиваются другие значения.

4. Составные операторы обозначаются с помощью вертикальных линий, которые добавляются путем щелчка мышкой по оператору **add**. Для добавления линии внутри текста программы следует полностью выделить строку, после которой следует вставить новую строку, и выполнить команду **add**. Выделение осуществляется нажатием клавиши «пробел», вертикальная линия выделения должна находиться в конце строки. Аналогично можно вставить новую строку перед выделенной строкой, только вертикальная линия уголка должна располагаться перед этой строкой.

5. В арифметических выражениях используется специфический знак присваивания: стрелка справа налево (\leftarrow).

6. В логических выражениях и для организации циклов используются:

а) *равно* – **жирный** знак равенства;

б) *больше или равно*;

в) *меньше или равно*;

г) *больше* (ввод клавишей >);

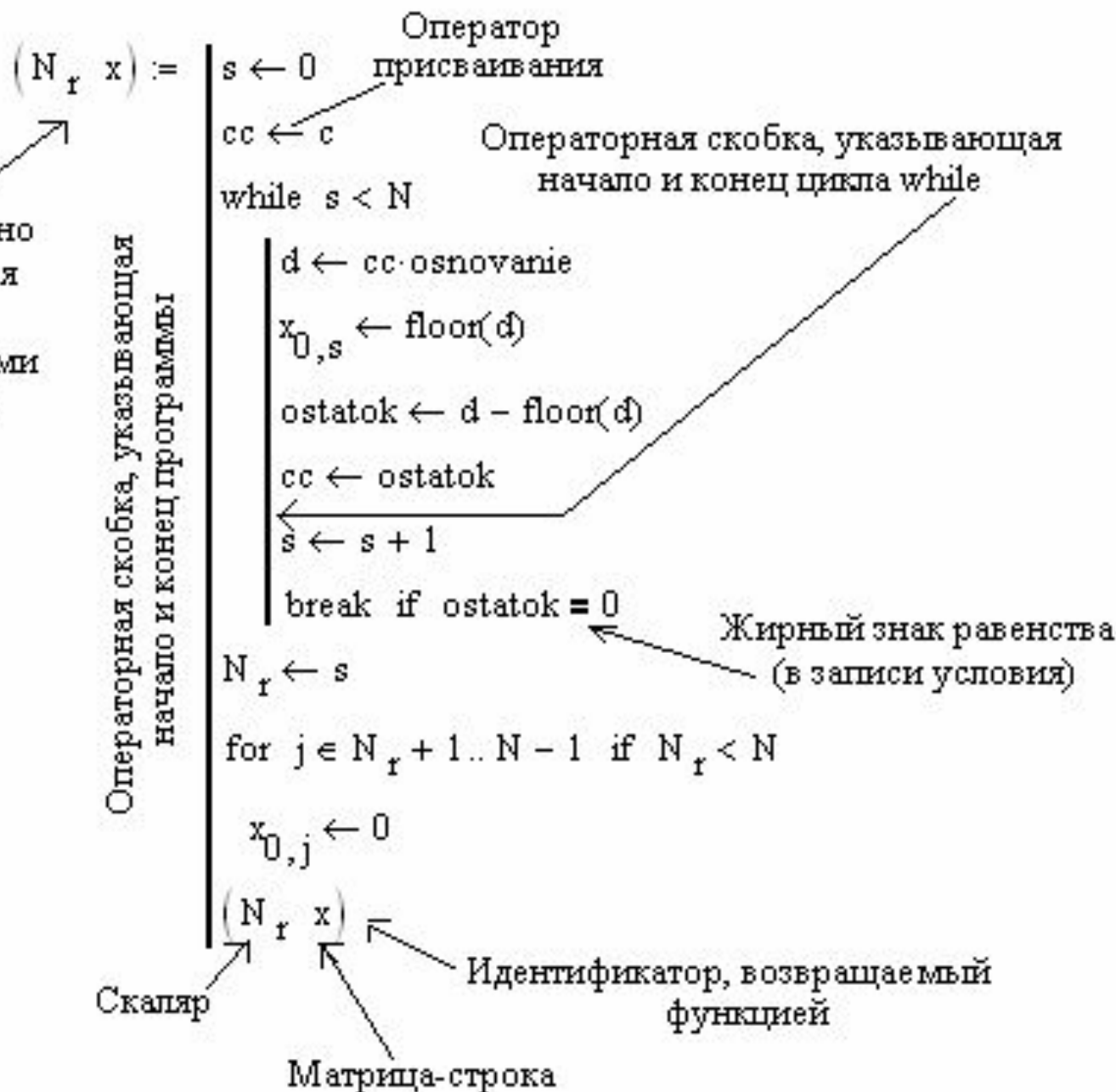
д) *меньше* (ввод клавишей <);

е) знаки логических действий «И» и «ИЛИ» (ввод знаком умножения (*) и сложения (+)).

7. Функция должна заканчиваться идентификатором возвращаемых переменных (простая переменная, матрица или структура). Запись оператора **return** не обязательна.

8. Начало программного блока содержит идентификатор результата, которому присваивается значение, вычисленное в блоке. Как показано в примере программного блока для возврата двух значений используется вектор из двух компонентов и в программном блоке, и в описании результата (объявлены как матрица из одной строки и двух столбцов). В свою очередь, один из компонентов представляет собой скалярную величину, а другой – вектор.

Ответ. Можно
пользоваться
отдельно
переменными
 N_r и x



10. Никакие знаки окончания операторов не ставятся. Если в операторе if при выполнении условия необходимо выполнить несколько операторов, то следует выделить в этом операторе левый шаблон ■ для заполнения и добавить необходимое количество строк обычным образом. При наличии нескольких операторов, выполняемых при соблюдении условия, структура оператора if меняется, соответствующие строки автоматически помещаются под оператором.

$$(w1 \ r) := \left| \begin{array}{l} i \leftarrow 0 \\ \text{if } 0 < x < 1 \\ \quad \left| \begin{array}{l} x \leftarrow x + 0.3 \\ i \leftarrow i + 1 \end{array} \right. \\ (i \ x) \end{array} \right.$$

3. Анимация графиков.

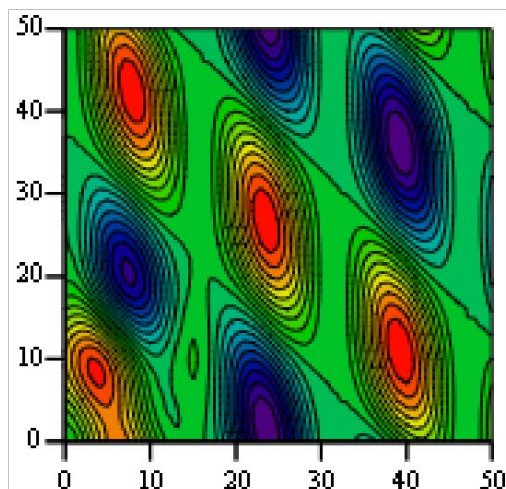
Анимация графиков позволяет наглядно отобразить влияние изменения какого-либо параметра на поведение функции. Пакет Mathcad содержит очень простые по составу и возможностям средства анимации любых типов графиков. Подготовка графика к анимации:

- задание некоторой переменной типа FRAME как ранжированной;
- задание функции для построения обычного графика. В описание функции следует включить переменную типа FRAME, которая определяет направление и динамику кадров;
- построить график и настроить его параметры;
- выполнить команду Tools – Animation – Record...;
- в появившемся окне задать параметры анимации;
- в окне документа выделить требуемое изображение рамкой из пунктирных линий;
- запустить анимацию, щелкнув по кнопке Animation. По результатам просмотра и при желании уточнить параметры анимации;
- сохранить анимацию в отдельном файле.

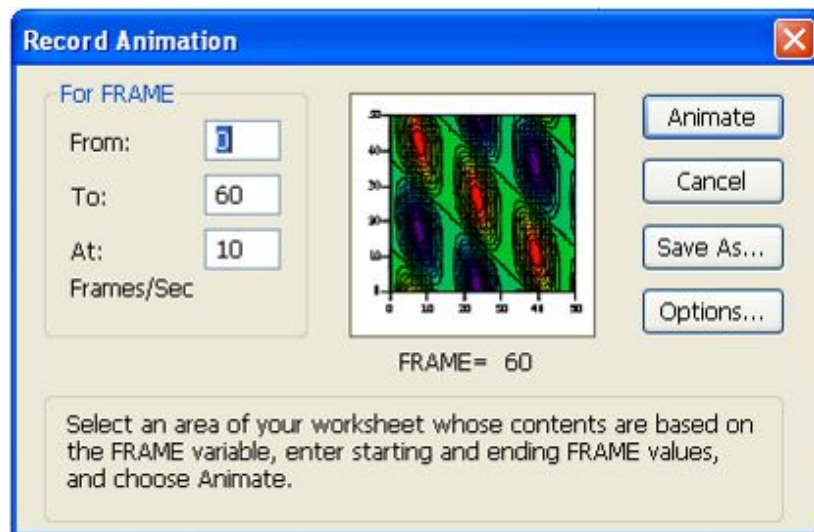
Пример

$g := \text{FRAME}$ $i := 0..50$ $j := 0..50$

$$a_{i,j} := \sin\left(\frac{i}{5}\right)^2 \cdot \cos\left(\frac{i+j}{8}\right) + \exp\left[\frac{-(i-5-g)^2 - (j-10-g)^2}{50}\right]$$



a



Воспроизведение анимации:

- выполнить команду Tools – Animation – Playback.... Появится окно Play Animation. В окне с помощью кнопки Open... открыть окно для выбора сохраненного файла анимации. Выбрать файл для отображения;