

Система Mathcad: файлы **данных** и функции доступа к ним

Лекция №8

Не путать: с возможностями
пункта File главного меню по
работе с документами Mathcad!

Возможности Mathcad по работе с файлами данных

Работа с текстовыми файлами числовых данных в формате ASCII

Импорт-экспорт файлов других приложений:
Excel, Matlab,
различных СУБД,
блокнота, ...

Работа с графическими, звуковыми файлами

Работа с текстовыми файлами ЧИСЛОВЫХ ДАННЫХ

Числа в файлах:

- целые: -1 34 -59
- вещественные, представленные во внешнем формате с фиксированной точкой: 97.73
- вещественные, представленные в экспоненциальном формате (внешнем формате с плавающей точкой):
4.51 E-4 -56.1 E5

*внутреннее
представление
вещественных данных – с
плавающей точкой!*

Разделители чисел в текстовых файлах

- В вводимых файлах: запятая, пробелы, перевод строки.
- В выводимых файлах: пробелы, перевод строки.
- Пример допустимой строки вводимого файла:
200, 50 25.1256, 16E -2, -16.125E15

Функции доступа к **неструктурированным** файлам данных

READ (<i>file</i>)	Считывает значение из файла данных. Возвращает скаляр . Обычно используется следующим образом: $v_i := \text{READ}(file)$
WRITE (<i>file</i>)	Записывает значение (одно!) в файл данных. Если файл уже существует, заменяет его на новый файл. Должна использоваться в операторах следующего вида: $\text{WRITE}(file) := v_i$
APPEND (<i>file</i>)	Дописывает значение (одно!) к существующему файлу. Должна использоваться в операторах следующего вида: $\text{APPEND}(file) := v_i$

Неструктурированные файлы – файлы, числа в которых необязательно расположены в строках и столбцах.

Структурированные файлы – числа в которых расположены в строках и столбцах, т. е. файлы, содержащие матрицу.

Неструктурированные файлы могут использоваться с индексированными переменные, структурированные – нет.

:= имеет смысл «считай, загрузи, введи» или «запиши, выведи, сохрани»

Функции доступа к **структурированным** файлам данных

READPRN (<i>file</i>)	Читает структурированный файл данных. Возвращает матрицу. Каждая строка в файле данных становится строкой в матрице. Число элементов в каждой строке должно быть одинаковым. Обычно используется следующим образом: A := READPRN(file)
WRITEPRN (<i>file</i>)	Записывает матрицу в файл данных. Каждая строка матрицы становится строкой в файле. Должна использоваться в определениях следующего вида: WRITEPRN(file) := A
APPENDPRN (<i>file</i>)	Дописывает матрицу к существующему файлу. Каждая строка в матрице становится новой строкой в файле данных. Должна использоваться в определениях следующего вида: APPENDPRN(file) := A. Существующий файл должен иметь столько же столбцов, как и матрица A.

Аргумент функций доступа к файлам данных

1. Аргумент файл называется файловой переменной и представляет собой имя файла (без кавычек), с расширением или без него.
2. Если расширение файла не указано, то для неструктурированных файлов оно по умолчанию считается .dat, для структурированных – .prn. Эти расширения являются предпочтительными.
3. Можно использовать расширения, отличные от .dat и .prn, но при вводе эти расширения записываются как нижний индекс. Например, вместо A.txt получим A_{txt} .
4. Не рекомендуется использовать расширение более чем из трех символов, иначе система будет искать файл по расширению из первых трех символов.
5. Файлы должны находиться только в текущей папке, т. е. в той папке, где находится документ Mathcad. Т. е. полное имя файла с указанием пути к нему в качестве файловой переменной использовать нельзя. При необходимости использовать файлы из других папок следует связать файловую переменную с файлом на диске с помощью окна Главное меню/ Файл /Присоединить к файлу.

Особенности использования функций доступа к файлам данных

- Если Mathcad не может найти файл данных, он отмечает соответствующую функцию доступа сообщением об ошибке “*файл не найден*”. Если Mathcad пытается прочитать файл неподходящего формата, он отмечает функцию сообщением “*ошибка файла*”.
- Каждое новое равенство с использованием функций доступа заново открывает файл данных. При считывании данных, например, каждое новое равенство начинает читать файл с самого начала.
- В одном равенстве файл может быть открыт только единожды. Это означает, что, если функция *READ* используется с одним аргументом-именем файла дважды в одном операторе (это возможно при использовании индексированной переменной), во второй раз *READ* начнет читать с того места, где закончила в первый. Поскольку *READPRN* читает весь файл целиком, это означает, что *READPRN* нельзя использовать с одним аргументом дважды в одном равенстве (в цикле) — во второй раз *READPRN* не останется ничего читать.
- Если два равенства в рабочем документе используют *WRITE* или *WRITEPRN* с одним аргументом, данные из второго равенства запишутся поверх данных из первого. Используйте *APPEND* или *APPENDPRN*, если нужно сохранить первую порцию данных. Эти функции дописывают новые данные к существующему файлу.

Примеры чтения данных из неструктурированного файла

Пример 1. В массив y из файла `in.dat` считывается 31 значение:

$N:=30$

$i:=0..N$

$y_i:=\text{READ}(\text{in})$

Пример 2.

$N:=\text{READ}(\text{sizefile})$ из файла `sizefile.dat` считывается значение N

$N=100$

$i:=1..N$ из файла `data.dat` считывается N значений в

$x_i:=\text{READ}(\text{data})$ массив x

Пример записи данных в неструктурированный файл

$N := 100$ $a := 1.75$ $b := 115$

$\text{WRITE}(\text{sizefile}) := N$

$i := 0..N-1$

$y_i := a \cdot i + b + \text{rnd}(i)$

$\text{WRITE}(\text{data1}) := y_i$

$\text{rnd}(i)$ – функция, возвращает псевдослучайное число, равномерно распределенное на интервале $[0, i]$

Пример работы со структурированным файлом

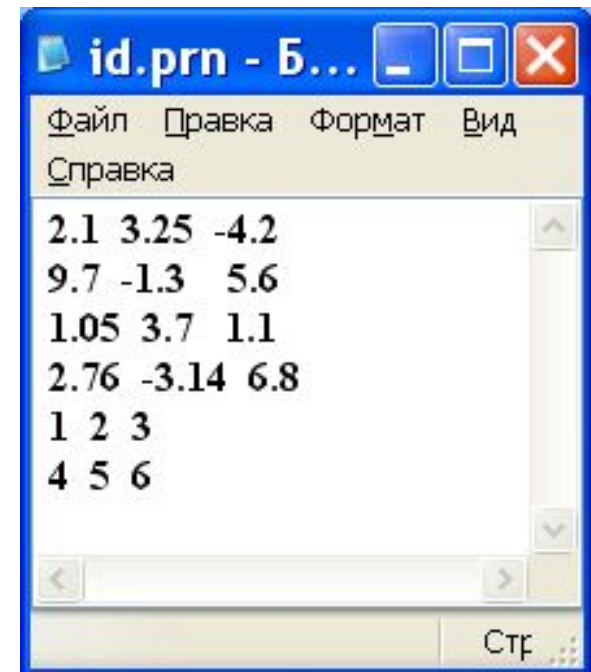
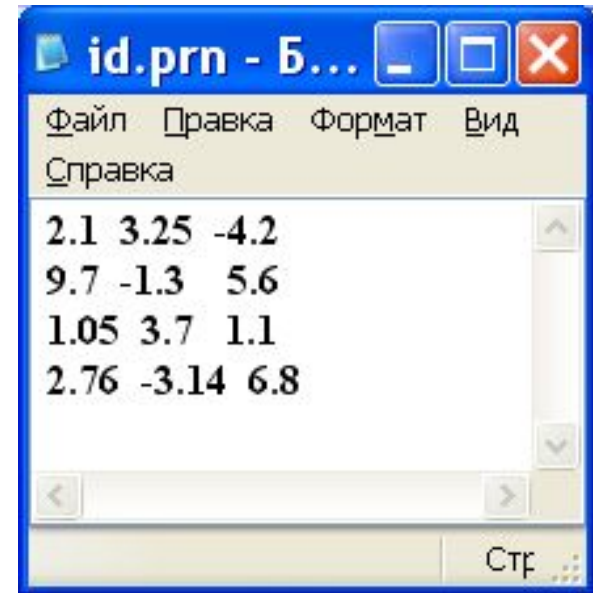
В блокноте набрали файл id.prn

$M := \text{READPRN}(\text{id})$

$$M = \begin{pmatrix} 2.1 & 3.25 & -4.2 \\ 9.7 & -1.3 & 5.6 \\ 1.05 & 3.7 & 1.1 \\ 2.76 & -3.14 & 6.8 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

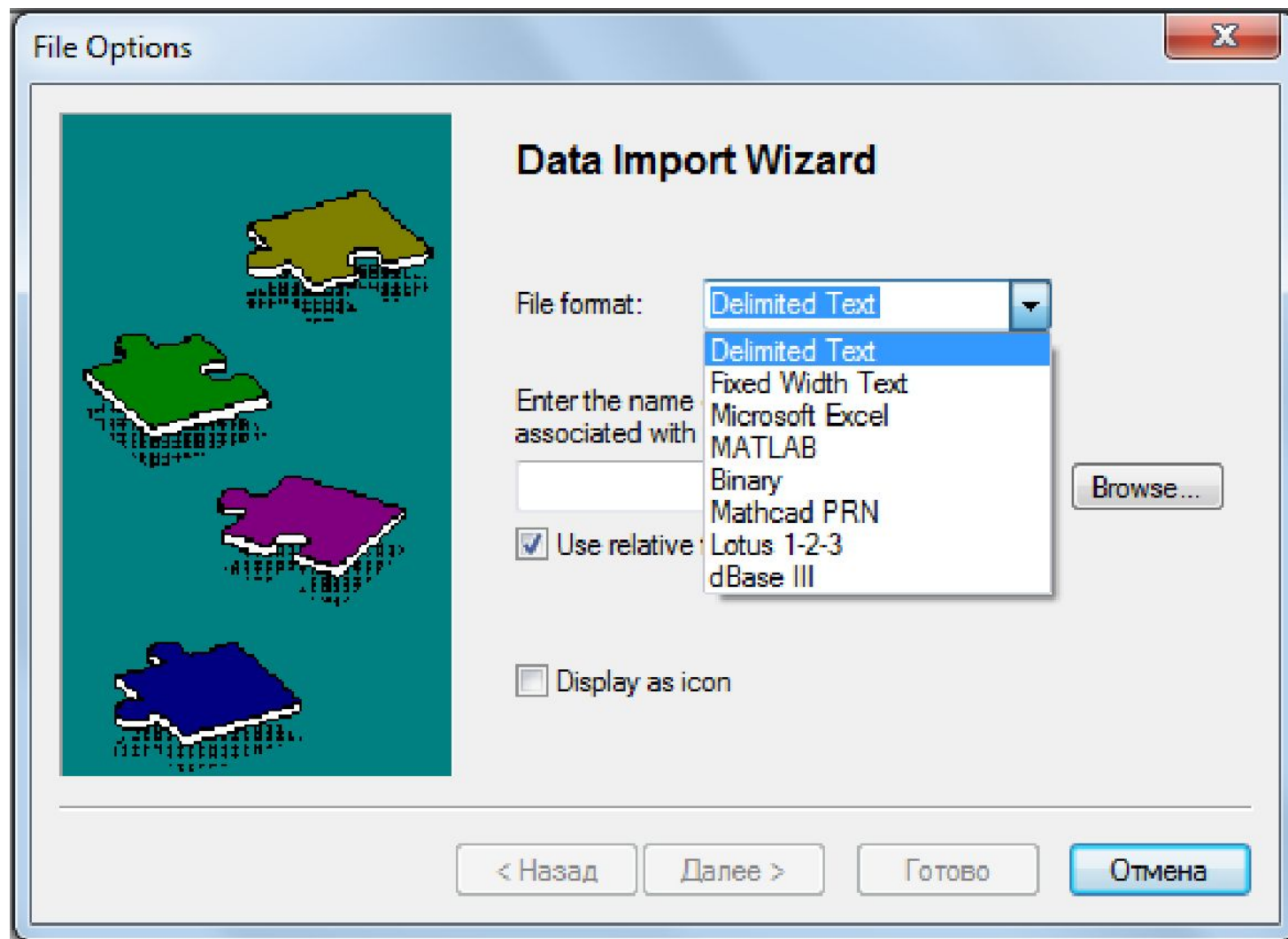
$\text{APPENDPRN}(\text{id}) := C$



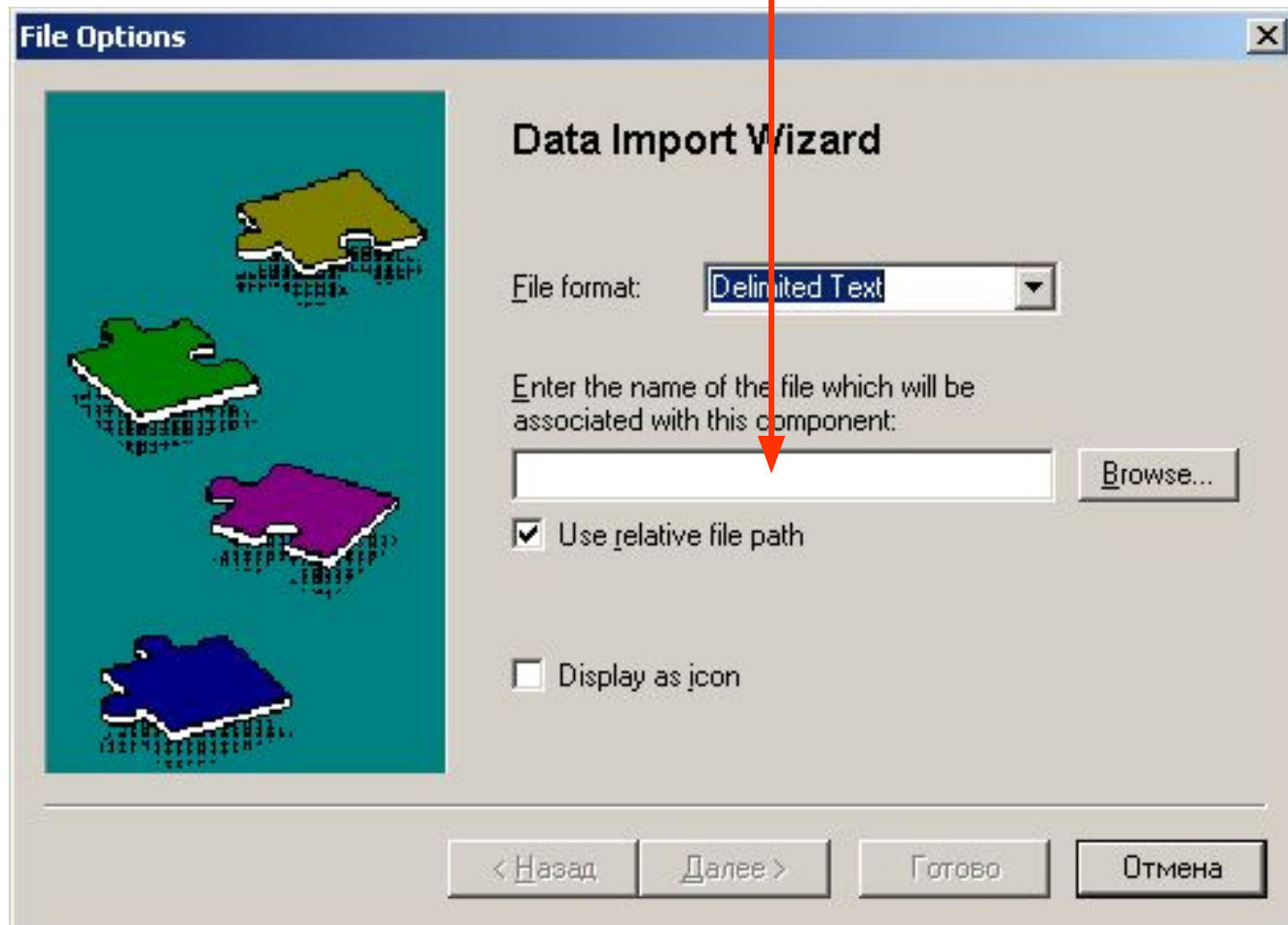
Импорт файлов, созданных различными приложениями

Главное меню/ Вставка/ Данные

1. Выбираем формат файла



2. Выбираем имя файла



3. Просматриваем и настраиваем файл

Delimited Text Options

Delimiter

Auto Comma

Tab Other:

Space

Starting row:

Blank rows:

Footer rows:

Preview

	1	2	3	4	5	
1	1	0	0	0	0	
2	0	1	0	0	0	
3	0	0	1	0	0	
4	0	0	0	1	99	
5	0	0	0	0	1	
6	1	2	3	4	5	

< Назад Далее > **Готово** Отмена

Результат в документе Mathcad

The screenshot displays the Mathcad software interface. At the top, there is a toolbar with various mathematical symbols and functions. Below the toolbar, the variable A is defined as a 6x6 matrix. The matrix is shown in a grid format with columns labeled 0 through 4 and rows labeled 0 through 5. Below the grid, the matrix is also represented symbolically as $A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 99 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$. The variable x is also present in the interface.

	0	1	2	3	4
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	99
4	0	0	0	0	1
5	1	2	3	4	5

$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 99 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

x

Press F1 for help. AUTO NUM Page 1

В подготовленное место вводим имя переменной

Работа с графическими файлами

`READRGB("file")` — чтение цветного изображения;
`READBMP("file")` — чтение изображения в оттенках серого;
`WRITERGB("file")` — запись цветного изображения;
`WRITEBMP("file")` — запись изображения в оттенках серого.

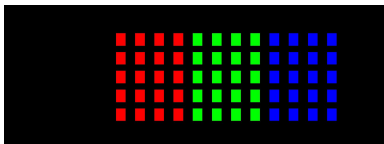
Существует также много функций преобразования файлов.

Форматы файлов: BMP, GIF, JPG, ...

Независимо от формата файла, изображение хранится как матрица кодов цветов пикселей изображения.

Изображение в оттенках серого: матрица, хранящая интенсивность серого от 0 до 255.

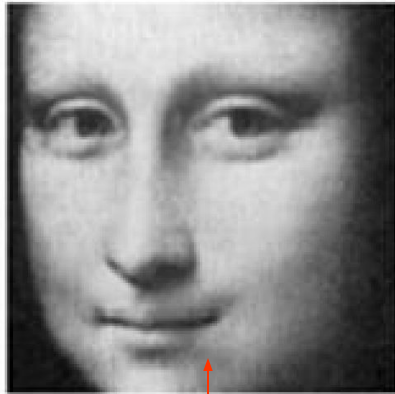
Для цветного изображения - три матрицы:



Пример работы с графическими файлами

```
M := READBMP("C:\Temp\Mona.bmp")
```

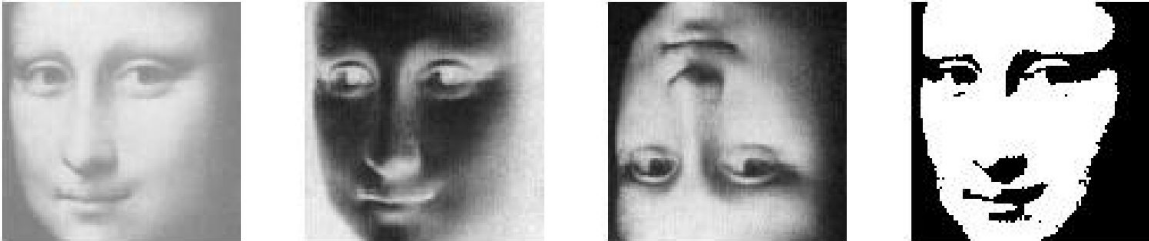
```
S := submatrix(M, 60, 160, 50, 150)
```



S

Вывод изображения командой Вставка/ Рисунок

Пример работы с графическими файлами (продолжение)

$$f(x) := \text{if}(x < 120, 0, 255)$$
$$S1 := \frac{S}{2} + 120 \quad S2 := 255 - S \quad S3 := \text{reverse}(S) \quad S4 := f(S)$$


S1 S2 S3 S4

1. Изменим уровень интенсивности серого оттенка в каждом пикселе, разделив матрицу S на 2 и добавив к результату 120. Получим матрицу $S1$.
2. Создадим инверсию графического изображения в матрице $S2$. Для этого значение интенсивности в каждом элементе матрицы $S2$ вычислим как $S2_{i,j} = 255 - S_{i,j}$.
3. Поменяем строки матрицы так, чтобы номера строк шли по убыванию с помощью функции $\text{reverse}()$ – матрица $S3$. Изображение должно перевернуться.
4. Проведем контрастирование изображения. Для этого все значения ниже 120 переназначим в 0, а выше в 255, т.е. оттенки будут только черные и белые. Получим матрицу $S4$.

Пример из книги: Ю.Е. ВОСКОБОЙНИКОВ, А.Ф. ЗАДОРОВЫЙ, Л.А. ЛИТВИНОВ, Ю. Г. ЧЕРНЫЙ «ОСНОВЫ ВЫЧИСЛЕНИЙ И ПРОГРАМИРОВАНИЯ В ПАКЕТЕ MATHCAD»

СПАСИБО ЗА ВНИМАНИЕ!