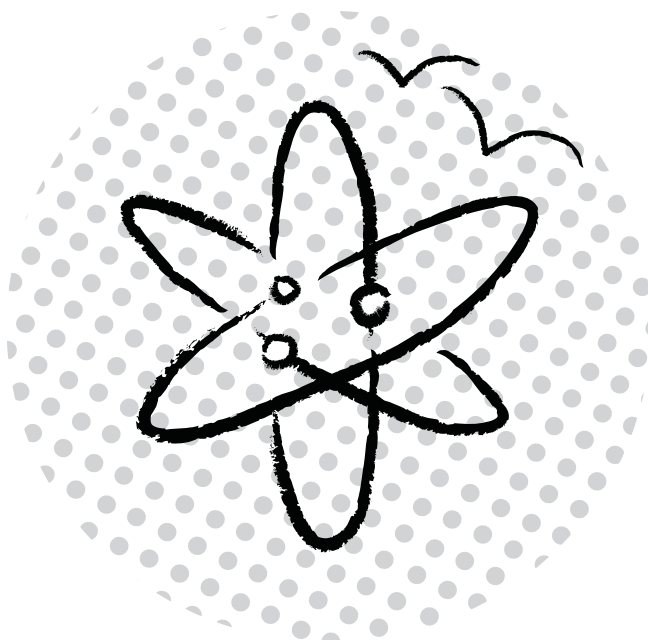


  
**OpenOffice.org** **pro**   
ИНФРА  
РЕСУРС

автоматизация работы



На CD OpenOffice.org pro и примеры программ



**УДК 004.4**  
**ББК 32.973.26018.2**  
**П32**

**П32** Эндрю Питоньяк (Andrew Pitonyak)  
OpenOffice.org pro. Автоматизация работы. М: ДМК Пресс, 2008.  
512 с., ил. + CD  
Перевод с английского — Заимских А. Н.

**ISBN 978-5-94074-441-2**

Эта книга — первое выходящее на русском языке издание по программированию в OpenOffice.org, в основе которого лежит статья Эндрю Питоньяка «Макросы».

Эндрю Питоньяк признанный мастер по написанию макросов. Его книги программисты разбирают на цитаты. Приведённые примеры с описанием позволяют сделать сложные вещи.

Теперь эта книга доступна и вам. На прилагаемом CD вы найдёте все листинги из этой книги, а так же OpenOffice.org pro, который совершенно легально можете установить у себя дома и на всех компьютерах в офисе.

УДК 004.4  
ББК 32.973.26018.2

**GNU FDL.** Все права защищены. Любая часть этой книги **МОЖЕТ** быть воспроизведена в любой форме и любыми средствами в любых целях без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок всё равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несёт ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-94074-441-2 © Текст, обложка «Инфра-Ресурс», 2008  
© Издание, «ДМК Пресс», 2008

## Подробное оглавление

От редакции.....	IX
От переводчика.....	IX
<b>Об этой книге.....</b>	<b>1</b>
Благодарности.....	3
Заявление.....	3
Предисловие автора.....	3
Контактная информация.....	4
Сокращения и обозначения.....	5
Доступные ссылки.....	6
Ссылки на сайты.....	7
<b>Программирование в ООо.....</b>	<b>9</b>
Глава I. Начало: концепция.....	11
I.1. Мой первый макрос: «Hello World».....	11
I.2. Группировка текста программ.....	12
I.3. Отладка.....	12
I.4. Переменные, константы, строки и числовые типы.....	12
I.5. Обращение к объектам и создание объектов (Objects) в OpenOffice.....	13
I.6. Что же такое UNO?.....	15
I.7. Итоги.....	24
Глава II. Примеры.....	25
II.1. Отладка и проверка макросов.....	25
II.2. Средство X-Ray.....	27
II.3. Диспетчер ООо (Dispatch): использование универсального сетевого объекта Universal Network Objects (UNO).....	28
II.4. перехват выполнения команд меню с использованием Basic.....	31
Глава III. Разные примеры.....	35
III.1.1. Вывод текста в строку состояния.....	35
III.2. Вывод всех стилей текущего документа.....	35
III.3. Перебрать все открытые документы.....	36
III.4. Вывести шрифты и другую информацию об экране.....	36
III.5. Установка шрифта по умолчанию с использованием ConfigurationProvider .....	38
III.6. Печать текущего документа.....	38
III.7. Информация о конфигурации.....	40
III.8. Открыть и закрыть документы (и рабочий стол).....	41
III.9. Загрузка / Вставка графики в документ.....	50
III.10. Установка полей.....	55
III.11. Установка размера бумаги.....	56
III.12. Вызов внешней программы (Internet Explorer) с помощью OLE.....	56
III.13. Использовать команду Shell для файлов, содержащих пробелы.....	57
III.14. Чтение и Запись числа в файл.....	57
III.15. Создать стиль числового формата.....	58
III.16. Посмотреть существующие числовые стили формата.....	59
III.17. Возвращает массив чисел Фибоначчи.....	60
III.18. Вставка текста в месте закладки (Bookmark).....	61
III.19. Сохранение и экспорт документа.....	61

III.20. Поля пользователя.....	63
III.21. Информация о документе.....	63
III.22. Текстовые поля.....	64
III.23. Мастер-поля (Master).....	65
III.24. Удаление текстовых полей.....	70
III.25. Вставка адреса URL в ячейку OOO Calc.....	70
III.26. Добавление текстового поля — формулы (SetExpression TextField).....	71
III.27. Типы данных, определённые пользователем.....	72
III.28. Проверка орфографии, переносы и толковый словарь — тезаурус.....	72
III.29. Изменение указателя мыши.....	74
III.30. Установка фона страницы (Page Background).....	75
III.31. Работа с буфером (clipboard).....	76
III.32. Установка локали (языка).....	81
III.33. Установки локали для выделенного текста.....	82
III.34. Автотекст (Auto Text).....	84
III.35. Десятичные футы и дроби.....	86
III.36. Преобразовать число в слова.....	90
III.37. Отправка электронного письма.....	94
III.38. Библиотеки макросов.....	97
III.39. Установка размеров точечного рисунка (Bitmap).....	101
III.40. Извлечение файла Zip.....	106
III.41. Выполнить макрос по строке с его именем.....	111
III.42. Использование «приложения по умолчанию» для открытия файла.....	113
III.43. Распечатка перечня шрифтов.....	113
III.44. Получить для документа: адрес URL, имя файла и папку.....	114
III.45. Получить и установить текущую папку (directory).....	114
III.46. Запись файла.....	117
III.47. Логический разбор синтаксиса (Parsing) XML.....	117
III.48. Работа с датами Dates.....	121
III.49. Встроен ли OpenOffice в Веб-браузер?.....	123
III.50. Активизировать (поставить на первый план) новый документ.....	123
III.51. Каков тип документа (основываясь на адресе URL).....	123
III.52. Соединиться с удаленным сервером OOO с использованием Basic.....	124
III.53. Панели инструментов.....	125
Глава IV. Макросы электронных таблиц Calc.....	133
IV.1. Является ли этот документ электронной таблицей?.....	133
IV.2. Вывести для ячейки электронной таблицы значение (value), строковое значение (string) и формулу (formula).....	133
IV.3. Установить для ячейки электронной таблицы значение (value), формат (format), текстовое значение (string) и формулу (formula).....	134
IV.4. Ссылка на ячейку в другом документе.....	134
IV.5. Очистка ячейки.....	134
IV.6. Выделенный (Selected) текст — что это?.....	135
IV.7. Простой пример обработки выделенных ячеек.....	136
IV.8. Получить активную ячейку игнорировать остальное.....	138
IV.9. Выделить (Select) ячейку.....	139
IV.10. Удобный для чтения (Human readable) адрес ячейки.....	140
IV.11. Вставить форматированную дату в ячейку.....	141

---

IV.12. Более короткий путь для этого.....	142
IV.13. Вывести выделенный интервал ячеек (selected range) в окно сообщения .....	142
IV.14. Заполнить выделенный интервал ячеек заданным текстом.....	143
IV.15. Некоторые данные и статистика о выделенном интервале ячеек.....	143
IV.16. (Именованный) Интервал ячеек базы данных.....	144
IV.17. Определить выбранные ячейки в качестве (именованного) интервала ячеек базы данных.....	145
IV.18. Удалить (именованный) интервал ячеек базы данных.....	146
IV.19. Границы таблицы.....	146
IV.20. Интервал ячеек для сортировки.....	147
IV.21. Вывести все данные столбца.....	149
IV.22. Использование методов объединения (Outline, Grouping).....	149
IV.23. Защита данных.....	150
IV.24. Установка текстов верхнего и нижнего заголовков.....	150
IV.25. Копирование листа целиком в новый документ.....	151
IV.26. Выделить именованный интервал ячеек (named range).....	152
IV.27. Выделить столбец целиком.....	154
IV.28. Выделить строку целиком.....	154
IV.29. Преобразовать данные из столбца определенного вида в строки.....	154
IV.30. Включить/Выключить автоматический пересчет.....	156
IV.31. Какие ячейки листа используются?.....	157
IV.32. Поиск в электронной таблице Calc.....	158
IV.33. Напечатать интервал ячеек электронной таблицы.....	161
IV.34. Объединена ли эта ячейка (с другими)?.....	162
IV.35. Написать свою функцию электронной таблицы Calc.....	162
Глава V. Макросы текстового документа (Writer) .....	167
V.1. Что такое выделенный текст?.....	167
V.2. Что такое текстовые курсоры (Text Cursors)?.....	169
V.3. Работа с текстом (Andrew's Selected Text Framework).....	173
V.4. Замена выделенного пробела с использованием строк (Strings). Вставка специальных символов.....	192
V.5. Установка атрибутов текста.....	203
V.6. Вставить текст.....	204
V.7. Поля.....	205
V.8. Вставка новой страницы.....	206
V.9. Установить стиль страницы в документе.....	207
V.10. Включение и выключение верхних и нижних заголовков.....	208
V.11. Вставить OLE-объект.....	208
V.12. Установка стиля параграфа (Paragraph).....	209
V.13. Создать свой собственный стиль.....	210
V.14. Поиск и замена.....	210
V.15. Изменение строчных букв на прописные и наоборот (Case) в словах.....	215
V.16. Перебор параграфов (поведение текстового курсора).....	217
V.17. Где находится курсор дисплея (Display Cursor)?.....	228
V.18. Вставка индекса или оглавления (table of contents).....	232
V.19. Вставка адреса URL в документ OOo Writer.....	233
V.20. Сортировка текста.....	233

V.21. Нумерация структур (Outline).....	234
V.22. Вставить оглавление (table of contents =TOC) или другой индекс.....	236
V.23. Текстовые секции (sections).....	237
V.24. Сноски на странице (Footnotes) и сноски в конце текста (Endnotes).....	239
Глава VI. Текстовые таблицы.....	241
VI.1. Поиск текстовых таблиц.....	241
VI.2. Перебор ячеек в текстовой таблице.....	244
VI.3. Извлечение данных из простой текстовой таблицы.....	250
VI.4. Курсоры таблицы и интервалы ячеек.....	251
VI.5. Интервалы ячеек (Cell ranges).....	251
VI.6. Данные диаграммы (Chart data).....	252
VI.7. Ширина столбцов.....	252
VI.8. Установка оптимальной ширины столбца.....	253
VI.9. Насколько широка текстовая таблица?.....	254
VI.10. Курсор в текстовой таблице.....	254
VI.11. Создание текстовой таблицы.....	257
VI.12. Таблица без рамок границы (borders).....	258
Глава VII. Форматирование макроса.....	259
VII.1. Утилиты для строк и массивов.....	259
VII.2. Утилиты для поиска разделов/секций с кодами макросов.....	266
VII.3. Основной модуль макроса.....	268
Глава VIII. Формы.....	275
VIII.1. Введение.....	275
VIII.2. Диалоги.....	276
VIII.3. Получение элементов управления (Controls).....	280
VIII.4. Выбор файла с использованием диалога (File Dialog).....	283
VIII.5. Центрировать диалог на экране.....	284
VIII.6. Установить перехватчик события (event listener) для элемента управления.....	284
VIII.7. Управление диалогом я не создал.....	286
Глава IX. Пример по инвестициям (Investment).....	301
IX.1. Внутренняя ставка возврата средств (Internal Rate of Return = IRR) .....	301
Глава X. Обработчики (Handlers) и перехватчики (Listeners).....	305
X.1. xKeyHandler пример.....	305
X.2. Перехватчик (Listener), описанный автором Paolo Mantovani.....	308
X.3. Что произошло с моим перехватчиком ActiveSheet?.....	317
Глава XI. Язык (Language).....	319
XI.1. Комментарии.....	319
XI.2. Переменные.....	319
XI.3. Функции (Functions) и процедуры (SubProcedures).....	332
XI.4. Управление последовательностью выполнения программы (Flow Control).....	335
XI.5. Разное.....	347
Глава XII. Операторы и их старшинство Operators and Precedence.....	349
Глава XIII. Действия со строками.....	351
XIII.1. Удалить символы из строки.....	352
XIII.2. Удалить текст из строки.....	353
XIII.3. Печать кодов ASCII символов строки.....	355

XIII.4. Удалить все экземпляры заданной подстроки из исходной строки.....	355
Глава XIV. Работа с числами.....	357
Глава XV. Работа с датами.....	359
Глава XVI. Работа с файлами.....	361
Глава XVII. Операторы в выражениях, операторы программы, функции.....	363
XVII.1. Оператор вычитания (-).....	363
XVII.2. Оператор умножения (*).....	363
XVII.3. Оператор сложения (+).....	364
XVII.4. Оператор возведения в степень (^).....	364
XVII.5. Оператор деления (/).....	365
XVII.6. Оператор AND .....	365
XVII.7. Функция Abs.....	366
XVII.8. Функция создания массива Array .....	367
XVII.9. Функция Asc.....	368
XVII.10. Функция ATN (арктангенс) .....	369
XVII.11. Оператор Beep.....	370
XVII.12. Функция Blue .....	370
XVII.13. Ключевое слово ByVal .....	371
XVII.14. Ключевое слово Call .....	371
XVII.15. Функция CBool .....	372
XVII.16. Функция CByte.....	373
XVII.17. Функция CDate .....	374
XVII.18. Функция CDateFromIso .....	374
XVII.19. Функция CDateToIso .....	375
XVII.20. Функция CDbt .....	375
XVII.21. Оператор ChDir — нежелателен.....	376
XVII.22. Оператор ChDrive — нежелателен.....	377
XVII.23. Функция Choose .....	377
XVII.24. Функция Chr .....	378
XVII.25. Функция CInt .....	379
XVII.26. Функция CLng .....	380
XVII.27. Оператор Close .....	380
XVII.28. Оператор константы Const .....	381
XVII.29. Функция ConvertFromURL .....	382
XVII.30. Функция ConvertToURL .....	382
XVII.31. Функция косинуса Cos .....	383
XVII.32. Функция создания диалога CreateUnoDialog .....	384
XVII.33. Функция CreateUnoService .....	385
XVII.34. Функция CreateUnoStruct .....	386
XVII.35. Функция CSng .....	386
XVII.36. Функция CStr .....	387
XVII.37. Функций CurDir .....	388
XVII.38. Функций Date .....	389
XVII.39. Функция DateSerial .....	389
XVII.40. Функция DateValue .....	390
XVII.41. Функция Day.....	391
XVII.42. Оператор объявления Declare .....	392
XVII.43. Оператор DefBool .....	393

XVII.44. Оператор DefDate .....	393
XVII.45. Оператор DefDbf .....	394
XVII.46. Оператор DefInt .....	394
XVII.47. DefLng Statement.....	395
XVII.48. Оператор DefObj .....	396
XVII.49. Оператор DefVar .....	396
XVII.50. Оператор Dim.....	397
XVII.51. Функция DimArray .....	398
XVII.52. Функция Dir.....	399
XVII.53. Операторы Do...Loop .....	401
XVII.54. Оператор End .....	402
XVII.55. Функция Environ .....	403
XVII.56. Функция EOF .....	403
XVII.57. Функция EqualUnoObjects .....	404
XVII.58. Оператор EQV .....	405
XVII.59. Функция Erl .....	406
XVII.60. Функция Err .....	406
XVII.61. Оператор Error не работает так, как описано.....	407
XVII.62. Функция Error .....	408
XVII.63. Оператор Exit .....	408
XVII.64. Функция Exp.....	410
XVII.65. Функция FileAttr .....	410
XVII.66. Оператор FileCopy .....	412
XVII.67. Функция FileDateTime .....	412
XVII.68. Функция FileExists .....	413
XVII.69. Функция FileLen .....	413
XVII.70. Функция FindObject .....	414
XVII.71. Функция FindPropertyObject .....	415
XVII.72. Функция Fix .....	416
XVII.73. Конструкция For...Next .....	417
XVII.74. Функция Format .....	417
XVII.75. Функция FreeFile .....	420
XVII.76. Функция FreeLibrary .....	421
XVII.77. Оператор Function .....	421
XVII.78. Оператор Get .....	422
XVII.79. Функция GetAttr .....	423
XVII.80. Функция GetProcessServiceManager .....	425
XVII.81. Функция GetSolarVersion .....	425
XVII.82. Функция GetSystemTicks .....	426
XVII.83. Ключевое слово GlobalScope .....	427
XVII.84. Оператор GoSub .....	427
XVII.85. Оператор GoTo .....	428
XVII.86. Функция Green .....	429
XVII.87. Функция HasUnoInterfaces .....	430
XVII.88. Функция Hex .....	431
XVII.89. Функция Hour.....	432
XVII.90. Оператор If .....	432
XVII.91. Функция IIF.....	433



XVII.92. Оператор Imp .....	434
XVII.93. Оператор Input .....	435
XVII.94. Функция InputBox .....	436
XVII.95. Функция InStr .....	437
XVII.96. Функция Int .....	438
XVII.97. Функция IsArray .....	439
XVII.98. Функция IsDate .....	440
XVII.99. Функция IsEmpty .....	440
XVII.100. Функция IsMissing .....	441
XVII.101. Функция IsNull .....	442
XVII.102. Функция IsNumeric .....	442
XVII.103. Функция IsObject .....	443
XVII.104. Функция IsUnoStruct .....	444
XVII.105. Функция Kill .....	444
XVII.106. Функция LBound .....	445
XVII.107. Функция LCase .....	446
XVII.108. Функция Left .....	446
XVII.109. Функция Len .....	447
XVII.110. Ключевое слово Let .....	447
XVII.111. Оператор Line Input .....	448
XVII.112. Функция Loc .....	448
XVII.113. Функция Lof .....	449
XVII.114. Функция Log .....	450
XVII.115. Оператор Loop .....	451
XVII.116. Оператор LSet .....	451
XVII.117. Функция LTrim .....	452
XVII.118. Ключевое слово Private .....	453
XVII.119. Ключевое слово Public .....	454
XVII.120. Функция Red .....	454
XVII.121. Оператор RSet .....	455
XVII.122. Функция Shell .....	456
XVII.123. Функция UBound .....	458
XVII.124. Функция UCase.....	459
XVII.125. Имена файлов и адреса URL.....	460
Глава XVIII. Другие языки.....	461
XVIII.1. С#.....	461
XVIII.2. Программисты на Visual Basic .....	462
<b>Приложения.....</b>	<b>465</b>
Приложение 1. Дополнения переводчика.....	467
Пример макроса, исправляющего оглавление. ....	467
Приложение 2. Горячие клавиши.....	469
Введение.....	469
Назначение сочетаний клавиш.....	469
Общие сочетания клавиш для OpenOffice.org.....	471
Сочетания клавиш, использующие функциональные клавиши.....	473
Сочетания клавиш в Галерее.....	474
Выделение строк и столбцов в таблице.....	476
Сочетания клавиш для рисованных объектов.....	476

Приложение 3. Особенности сборки OpenOffice.org pro.....	479
Общие улучшения.....	479
Улучшения в программе Writer.....	480
Улучшения в программе Calc.....	480
Улучшения в программе Impress .....	480
Улучшения в сборке для Windows.....	481
Улучшения в сборке для GNU/Linux и FreeBSD.....	481
Приложение 4. Алфавитный указатель (Index).....	483

## Глава I. Начало: концепция

**П**ервая глава моей книги, доступная для свободного скачивания, более полна для начинающего пользователя. Я считаю более полезным начать с неё:

<http://www.pitonyak.org/book/>

Макросы используются для автоматизации действий в OpenOffice.org. Макрос может автоматизировать такие действия, который иначе потребовали бы длительных ручных манипуляций с возможными ошибками. В настоящее время автоматизированные действия наиболее легко выполняются написанием макросов в OOo Basic. Новая среда для макросов в версии 2 OOo должна облегчить использование других языков, но Basic все ещё наиболее лёгкий в использовании. Вот несколько преимуществ использования языка OOo Basic для управления OOo:

- лёгок для изучения
- поддерживает объекты COM (ActiveX) и расширенные возможности GUI в OpenOffice
- есть сообщество пользователей в Интернет
- это — решение для нескольких платформ (Linux, Windows ...)

---

***Примечание** OpenOffice.org Basic известен также как StarBasic.*

### I.1. Мой первый макрос: «Hello World»

Откройте новый документ OOo . Используйте меню Сервис — Макросы — Управление макросами — OpenOffice.org Бэйсик, чтобы начать диалог макросов Macro. С левой стороны окна диалога найдите документ, который Вы только что открыли. Новый документ, вероятно, назван «untitled1» или Безымянный1. Кликните (нажмите левую клавишу мыши) справа ниже от «untitled1» на слове «standard». Кликните кнопку Создать далеко справа для создания нового модуля.

Использование имени «Module1», вероятно, не лучшее решение. Когда у Вас открыто несколько документов и все они имеют модуль с именем «Module1», то становится трудно работать с ними. Лучше назовём Ваш первый модуль «MyFirstModule». Откроется среда редактирования и отладки макросов OOo Basic IDE . Введите (или скопируйте) текст, приведённый в Лист. 1.

*Лист. 1: Ваш первый макрос, «Hello world».*

```
Sub Main
  Print "Hello World"
End Sub
```

Кликните на кнопке с зелёным треугольничком («Выполнить Basic») в верхней панели для выполнения Вашего первого макроса OOo Basic.

## I.2. Группировка текста программ

OOo Basic основан на процедурах и функциях, который задаются ключевыми словами Sub и Function — далее они будут называться процедурами (procedures, routines, subroutines) или соответственно функциями. Каждая процедура может вызывать другие процедуры. Разница между Sub и Function в том, что функция возвращает значение, а процедура — нет. Макрос на Лист. 2 получает текстовую строку от функции с именем HelloWorldString.

*Лист. 2: «Hello world» с использованием процедуры и функции.*

```
Sub HelloWorld
  Dim s As String
  s = HelloWorldString()
  MsgBox s
End Sub

Function HelloWorldString() As String
  HelloWorldString = "Hello World"
End Function
```

Каждый модуль (module) содержит набор процедур (функций). Библиотека (Library) содержит набор модулей. Документ (document) может содержать библиотеку или несколько библиотек. Библиотека может существовать также на уровне Приложения (application level), такого как OOo Writer.

## I.3. Отладка

Объединённая среда разработки IDE имеет средства отладки, такие как установка точек прерывания и наблюдаемых переменных. Вы можете также выполнять макрос по шагам, по оператору за один раз. Полезно установить точки прерывания перед подозрительным местом макроса и затем выполнять его по шагам, чтобы увидеть, как возникает ошибка.

## I.4. Переменные, константы, строки и числовые типы

Переменная похожа на коробку, которая что-то содержит. Как коробка, некоторые переменные больше подходят для определённых типов содержимого. Тип переменной определяет, что она может содержать. Попытка сохранить в ней данные неверного типа часто вызывают ошибку, конкретнее — Exception. До использования переменной она должна получить имя. Простой пример:

*Лист. 3: Определение простой переменной.*

```
Dim <variablename> as <Type>  
Dim i as integer
```

Переменные меняют свои значения, без этого Вы не можете сохранять данные. Как и содержимое коробки, содержимое переменной может меняться. У меня в комнате есть коробка, и только я могу изменить её содержимое. Теперь, я женился, так что моя жена тоже может менять содержимое коробки. Точнее говоря, область видимости переменной определяет, где доступна эта переменная. Вы можете определить переменную как доступную только для одной процедуры, или для целого модуля, или даже как глобальную — для всех библиотек. Ключевые слова «Global» (глобальная — для всех библиотек), «Public» (общая — для целого модуля), and «Private» (частная — только для одной процедуры или функции) используются для определения области видимости переменной. Местонахождение Вашего макроса, в котором определена эта переменная, также может изменять её область видимости.

Переменная, чьё значение не может быть изменено, называется константой (Constant). Константа определяется с использованием ключевого слова Const.

*Лист. 4: Определение константы.*

```
Const <constantname> = <constantvalue>  
const pi=3.14159265358
```

OOo Basic поддерживает много различных типов данных. Строки — это простые текстовые значения, ограниченные двойными кавычками. Что касается чисел, то поддерживаются и целые числа, и числа с плавающей точкой. Подробнее о переменных см. главу XI «Язык (Language)» на стр. 319 а также справку OOo. Можно прочесть и мою книгу «OpenOffice.org Macros Explained».

## **1.5. Обращение к объектам и создание объектов (Objects) в OpenOffice**

OpenOffice.org включает большое число сервисов и объектов (Services, Objects); обычно объекты легко доступны. Доступ к текущему документу и к рабочему столу использует глобальные переменные ThisComponent и StarDesktop соответственно — обе эти глобальные переменные являются объектами. Если у Вас есть документ, Вы можете получить доступ к нему (см. Лист. 5).

*Лист. 5: Определение и использование некоторых переменных.*

```
Sub Example
  Dim oDoc As Variant      ' Ссылка на активный документ
  Dim oText As Variant     ' Ссылка на главный объект Text
                          ' этого документа
  oDoc = ThisComponent    ' Получить активный документ
  oText = oDoc.getText()  ' Получить объект – текст
                          ' документа TextDocument
End Sub
```

Чтобы создать экземпляр объекта, используйте глобальный метод `createUnoService()` как показано в Лист. 6. Он также показывает, как создать структуру (structure).

*Лист. 6: Это старый способ отправки сообщений dispatch.*

```
Sub PerformDispatch(vObj, uno$)
  Dim vParser      ' Будет ссылаться на URLTransformer.
  Dim vDisp        ' Результат отправки сообщения.
  Dim oUrl As New com.sun.star.util.URL 'создаёт структуру

  oUrl.Complete = uno$
  vParser = createUnoService("com.sun.star.util.URLTransformer")
  vParser.parseStrict(oUrl)

  vDisp = vObj.queryDispatch(oUrl,"",0)
  If (Not IsNull(vDisp)) Then vDisp.dispatch(oUrl,noargs())
End Sub
```

---

**Совет** Руководство разработчика *Developer's Guide* гласит, что тип *Variant* должен использоваться вместо типа *object*. Это обсуждается подробнее на стр.326.

---

**Совет** Хотя Вы можете создать экземпляр объекта *Desktop*, но ниже будет показано, что Вам следовало бы вместо этого использовать глобальную переменную *StarDesktop*. Вам нужно создавать *Desktop* только на других языках, кроме *StarBasic*.

`createUnoService(«com.sun.star.frame.Desktop»)`

---

## I.6. Что же такое UNO?

UNO (Universal Network Object, Универсальный сетевой объект) был создан, чтобы каждая среда (окружение) environment могла успешно взаимодействовать с каждой другой средой (окружением). Почему это нужно? Потому что разные языки программирования и разные среды (окружения) могут иметь разные способы представления одного и того же типа данных. Даже целые числа, наиболее простая вещь, могут быть представлены разными способами на разных компьютерах и разных языках программирования.

- UNO определяет множество базовых типов, таких как строки, целые числа и т. д. (поэтому они будут одними и теми же в разном окружении).
- объекты UNO могут иметь методы (метод может возвращать значение и может принимать аргументы).
- объекты UNO могут иметь свойства. Свойство может быть другим объектом UNO или может иметь простой тип. Свойство может также быть необязательным (опциональным).
- объекты UNO определены с использованием сложного (для большинства читателей) Языка Определения Интерфейса Interface Definition Language (UNO IDL). Хотя Вам может не понравиться изучение UNO IDL, именно он определяет свойства и методы, которые поддерживают объекты.
- Предположим, что я имею окружение UNO в языке Basic, и другое окружение в языке Java. Для того, чтобы окружение UNO использовало объект, все, что требуется — определение объекта на языке UNOIDL. Эти окружения могут легко передавать объекты туда и обратно.

Я использую OOo API для взаимодействия с OOo. Эта технология API может возвращать:

- внутренний тип UNO такой как число с плавающей точкой, целое число и т. д.
  - константы — значения, обычно числовые, связанные с именем константы. Например, обычный текст имеет размер шрифта, определяемый константой `com.sun.star.awt.FontWeight.NORMAL`, которая равна 100.0.
  - нумерации, использующие имя похожее на `com.sun.star.awt.FontSlant.ITALIC`, но они обычно связаны с целым числом, даже если значение нумератора не выводится на печать.
-

- Структуры — это объекты со свойствами, но не имеющие методов. Сам по себе факт, что объект не имеет методов, не обязательно означает, что это структура. Объект должен быть именно определён как структура.
- UNO объект с методами и (или) свойствами.

### I.6.1. Структуры

Структуры, которые определены с помощью OpenOffice.org, могут быть проверены с использованием метода IsUNOStruct().

*Лист. 7: Проверка на структуру UNO.*

```
Sub ExamineStructures
  Dim oProperty As New com.sun.star.beans.PropertyValue
  With oProperty
    .name = "Joe"
    .value = 17
  End With
  Print oProperty.Name & " is " & oProperty.Value
  If IsUNOStruct(oProperty) Then
    Print "oProperty is an UNO Structure"
  End If
End Sub
```

Хотя Вы можете определить и использовать свои структуры, IsUNOStruct() не распознает их как структуры.

### I.6.2. Интерфейсы

Почти все объекты ООо поддерживают сервисы (службы, services) и интерфейсы (interface). Когда здесь используется слово «интерфейс», связанное с каким-либо объектом, то это слово всегда означает набор методов, которые этот объект поддерживает. Например, если объект поддерживает интерфейс com.sun.star.frame.XStorable или короче XStorable, то объект поддерживает методы, перечисленные в Таблице 2.

*Таблица 2. Методы, определённые интерфейсом XStorable.*

<b>Метод</b>	<b>Описание</b>
hasLocation	Возвращает true если объект имеет сведения о его расположении, либо потому, что он был загружен оттуда, либо потому, что он был сохранен туда.
getLocation	Возвращает адрес URL , куда объект был сохранен.
isReadOnly	<i>(только для чтения)</i> Если возвращает true, нельзя вызывать метод store().
store	Сохраняет данные по адресу URL , откуда они были



<b>Метод</b>	<b>Описание</b>
	загружены.
storeAsURL	Сохраняет объект по указанному адресу URL. Последующие вызовы метода store будут использовать этот адрес URL.
storeToURL	Сохраняет объекта по указанному адресу URL, но это не меняет адрес URL документа.

Макрос в Лист. 41 проверяет, поддерживает ли компонент интерфейс XStorable. Если да, то макрос использует методы `hasLocation()` и `getLocation()`. Проверка необходима потому, что, возможно, возвращаемый компонент не будет поддерживать интерфейс XStorable или данный документ не был сохранен и поэтому не имеет сведений о расположении, которые могли бы быть напечатаны.

### **I.6.3. Сервисы (Services)**

Когда здесь используется слово «сервис» как связанное с объектом, то это слово означает набор интерфейсов, свойств и сервисов, которые поддерживает объект. Для конкретного интерфейса или сервиса можно найти его определение по ссылке:

<http://api.openoffice.org>.

Определение сервиса задаёт только объекты, непосредственно определённые этим сервисом. Например, сервис `TextRange` определён как поддерживающий сервис `CharacterProperties`, который определён как поддерживающий конкретные свойства, такие как `CharFontName`. Объект, которых поддерживает сервис `TextRange` будет поэтому поддерживать и свойство `CharFontName`, даже если оно не указано непосредственно в определении объекта; Вам нужно просмотреть все перечисленные интерфейсы, сервисы и свойства, чтобы получить все, что реально поддерживает данный объект.

### **I.6.4. Интерфейсы и сервисы (services)**

Из вышесказанного следует, что, хотя доступны определения почти всех сервисов и интерфейсов, это не лучший способ быстро получить все сведения об объекте. BASIC IDE позволяет Вам поставить точки останова в Вашем коде программы для просмотра значений переменных. Многие программисты просматривают объекты, используя бесплатную библиотеку просмотра объектов, называемую `Xray`. У меня есть мои собственные процедуры просмотра объектов, как написано в моей книге `OpenOffice.org Macros Explained`.

Если объект содержит интерфейс XserviceInfo, то Вы можете узнать, поддерживает ли объект конкретный сервис:

<http://api.openoffice.org/docs/common/ref/com/sun/star/lang/XServiceInfo.html>

Эта возможность обычно используется для выяснения, принадлежит ли документ к конкретному виду документов.

*Лист. 8: Проверка, является ли документ текстовым документом.*

```
Dim s As String
s = "com.sun.star.text.TextDocument"
If ThisComponent.supportsService(s) Then
    Print "Документ является текстовым документом"
Else
    Print "Этот документ не является текстовым документом"
End If
```

### 1.6.5. Каков тип данного объекта?

Бывает полезно знать тип объекта, чтобы понять, что можно с ним делать. Вот краткий список способов проверки:

*Таблица 3: Способы, используемые для проверки переменных.*

<b>Method</b>	<b>Description</b>
IsArray	Является параметр массивом?
IsEmpty	Является параметр не инициализированной переменной типа variant?
IsNull	Является ли значение параметра пустым?
IsObject	Является параметр объектом OLE?
IsUnoStruct	Является параметр структурой UNO?
TypeName	Каково наименование типа параметра?

Имя типа переменной может также давать информацию о свойствах переменной.

*Таблица 4: Значения, возвращаемые выражением TypeName()*

<b>Тип</b>	<b>TypeName()</b>
Variant	«Empty» или имя содержащегося объекта
Object	«Object», даже если значение пустое. То же касается структуры
regular type	Имя обычного типа, такое как «String» (строка)
array	имя, заканчивающееся круглыми скобками «()»

Лист. 9 иллюстрирует выражения в Таблица 3,

*Лист. 9: Проверка переменных.*

```

Sub TypeTest
  Dim oSFA
  Dim aProperty As New com.sun.star.beans.Property
  oSFA = CreateUnoService( "com.sun.star.ucb.SimpleFileAccess" )
  Dim v, o As Object, s As String, ss$, a(4) As String
  ss = "Empty Variant: " & GetSomeObjInfo(v) & chr(10) & _
    "Empty Object: " & GetSomeObjInfo(o) & chr(10) & _
    "Empty String: " & GetSomeObjInfo(s) & chr(10)
  v = 4
  ss = ss & "int Variant: " & GetSomeObjInfo(v) & chr(10)
  v = o
  ss = ss & "null obj Variant: " & GetSomeObjInfo(v) & chr(10) & _
    "struct: " & GetSomeObjInfo(aProperty) & chr(10) & _
    "service: " & GetSomeObjInfo(oSFA) & chr(10) & _
    "array: " & GetSomeObjInfo(a())
  MsgBox ss, 64, "Type Info"
End Sub

REM Функция возвращает основную информацию о типе полученного параметра
REM Она возвращает также размерности массива.
Function GetSomeObjInfo(vObj) As String
  Dim s As String
  s = "TypeName = " & TypeName(vObj) & CHR$(10) & _
    "VarType = " & VarType(vObj) & CHR$(10)
  If IsNull(vObj) Then
    s = s & "IsNull = True"
  ElseIf IsEmpty(vObj) Then
    s = s & "IsEmpty = True"
  Else
    If IsObject(vObj) Then
      On Local Error GoTo DebugNoSet
      s = s & "Implementation = " & _
        NotSafeGetImplementationName(vObj) & CHR$(10)
      DebugNoSet:
      On Local Error Goto 0
      s = s & "IsObject = True" & CHR$(10)
    End If
    If IsUnoStruct(vObj) Then s = s & "IsUnoStruct = True" & CHR$(10)
    If IsDate(vObj) Then s = s & "IsDate = True" & CHR$(10)
    If IsNumeric(vObj) Then s = s & "IsNumeric = True" & CHR$(10)
    If IsArray(vObj) Then
      On Local Error Goto DebugBoundsError:
      Dim i%, sTemp$
      s = s & "IsArray = True" & CHR$(10) & "range = ("
      Do While (i% >= 0)
        i% = i% + 1
        sTemp$ = LBound(vObj, i%) & " To " & UBound(vObj, i%)
        If i% > 1 Then s = s & ", "
        s = s & sTemp$
      Loop
      DebugBoundsError:
    End If
  End If
  s = s & CHR$(10)
  GetSomeObjInfo = s
End Function

```

```

    On Local Error Goto 0
    s = s & ")" & CHR$(10)
  End If
End If
GetSomeObjInfo = s
End Function

REM Функция помешает перехватчик ошибок, в котором можно получить
REM информацию о проблеме и в любом случае выдать хоть что-нибудь!
Function SafeGetImplementationName(vObj) As String
  On Local Error GoTo ThisErrorHere:
  SafeGetImplementationName = NotSafeGetImplementationName(vObj)
  Exit Function
ThisErrorHere:
  On Local Error GoTo 0
  SafeGetImplementationName = "**** Unknown ****"
End Function

REM Проблема в том, что данная функция вызывается, а параметр vObj
REM не поддерживает вызов getImplementationName(),
REM поэтому я получаю ошибку вида "Object variable not set"
REM при определении функции.
Function NotSafeGetImplementationName(vObj) As String
  NotSafeGetImplementationName = vObj.getImplementationName()
End Function

```

### 1.6.6. Какие методы, свойства, интерфейсы и сервисы поддерживаются?

И сервисы, и структуры имеют имя типа «Object». Для структуры нужно знать тип этой структуры, чтобы можно было найти поддерживаемые свойства — используйте Интернет-сайт API для проверки языка описания интерфейсов IDL. Объекты UNO, с другой стороны, обычно поддерживают ServiceInfo, который предоставляет информацию о сервисе. Метод объектов getImplementationName() возвращает полное имя (то есть со всеми вышестоящими объектами) данного объекта. Используйте полное имя объекта для поиска в Интернете или в Руководстве разработчика. Метод объекта getSupportedServiceNames() возвращает перечень всех интерфейсов, поддерживаемых объектом. Общий способ выяснить, что может делать данный объект, — это вызов следующих трёх методов:

*Лист. 10: Что может делать этот объект?*

MsgBox	vObj.dbg_methods	'Методы данного объекта
MsgBox	vObj.dbg_supportedInterfaces	'Интерфейсы, поддерживаемые этим объектом
MsgBox	vObj.dbg_properties	'Свойства данного объекта

ООо включает макросы для вывода отладочной информации. Наиболее часто используются макросы PrintdbgInfo(объект) и ShowArray(объект). Макрос WritedbglInfo(объект) вставляет отладочную информацию в открытый документ ООо Writer.

### I.6.7. Языки, отличные от Basic

StarBasic предоставляет множество возможностей, которые не дают другие языки программирования. В этом разделе упоминаются только некоторые из этих возможностей.

#### I.6.7.A. Метод CreateUnoService

Метод CreateUnoService() — это короткий путь вместо вызова менеджера глобального сервиса (global service manager) и затем вызова createInstance() для этого менеджера.

*Лист. 11: Вызов менеджера глобального сервиса процессов.*

```
oManager = GetProcessServiceManager()
oDesk = oManager.createInstance("com.sun.star.frame.Desktop")
```

В языке StarBasic этот процесс можно сделать в одну строку — если только Вам не нужно использовать createInstanceWithArguments().

*Лист. 12: CreateUnoService даёт более короткий код программы, чем использование менеджера сервиса процессов.*

```
oDesk = CreateUnoService("com.sun.star.frame.Desktop")
```

Другие языки, такие как Visual Basic, не поддерживают метод CreateUnoService().

*Лист. 13: Создание сервиса UNO в языке Visual Basic.*

```
'Visual Basic не поддерживает CreateUnoService().
'менеджер сервиса процессов всегда создается первым
REM   в Visual Basic.
Rem   Если OOo не запущен ещё, то он запустится сейчас
Set oManager = CreateObject("com.sun.star.ServiceManager")
Rem   Создаем объект desktop
Set oDesk = oManager.createInstance("com.sun.star.frame.Desktop")
```

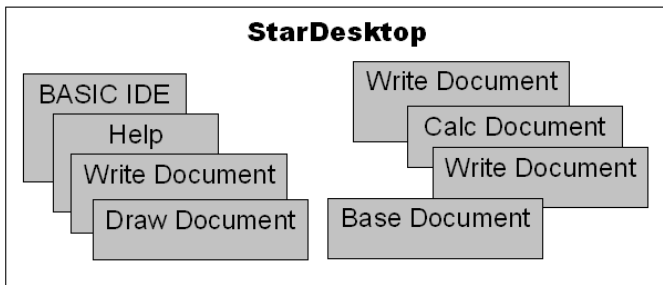
#### I.6.7.B. Глобальная переменная ThisComponent

В языке StarBasic, глобальная переменная ThisComponent ссылается на текущий документ или на документ, из которого был вызван данный макрос. Глобальная переменная ThisComponent получает своё значение один раз, когда макрос стартует, и затем не меняется, даже если этот макрос делает текущим другой документ — это включает закрытие ThisComponent. Даже со всеми проблемами ThisComponent очень удобен. В других языках, помимо Basic, общим способом является использование getCurrentComponent() для объекта рабочего стола (desktop). К сожалению, это не даст ссылку на объект документа, если текущим является окно Basic IDE или окно помощи, но это маловероятно, если используется другой язык.

Иногда Вам захочется использовать `StarDesktop.getCurrentComponent()` вместо того, чтобы просто использовать `ThisComponent`, но макрос выдаст ошибку, если он вызван из среды отладки Basic IDE.

### I.6.7.C. Глобальная переменная `StarDesktop`

Язык OOo Basic определяет несколько глобальных переменных и предоставляет их для удобства программиста. Переменная `StarDesktop` ссылается на объект рабочего стола (desktop object), который, по существу, является основным приложением OOo. Это имя появилось, когда продукт назывался StarOffice, и оно означало объект главного рабочего стола, который содержал все остальные компоненты. Примерами компонентов, которые может содержать объект рабочего стола, являются все поддерживаемые документы, интегрированная среда разработки BASIC Integrated Development Environment (IDE), а также страницы помощи и поддержки (см. Рисунок 2).



**Рисунок 2:** Компоненты Рабочего стола StarDesktop

Возвращаясь к макросу из Лист. 13, объект `StarDesktop` предоставляет доступ ко всем открытым в данный момент компонентам. Метод `getCurrentComponent()` возвращает компонент, активный в данный момент. Если макрос выполняется из BASIC IDE, то возвращается ссылка на BASIC IDE. Если макрос выполняется в то время, когда на экран остаётся выведен документ, вероятно выполняется посредством меню Сервис — Макросы — Выполнить, то `oComp` даст ссылку на текущий документ.

**Совет** Глобальная переменная `ThisComponent` ссылается на документ, активный в данный момент. Если активным является компонент другого типа, не документ, то `ThisComponent` ссылается на последний активный документ. В версии OOo 2.01 компоненты Basic IDE, страницы помощи, базы

данных (*Base documents*) не вызывают изменения *ThisComponent* на ссылку на такой компонент.

### I.6.8. Методы и свойства доступа к объектам в зависимости от языка

StarBasic автоматически делает методы и свойства, поддерживаемые объектом, доступными, причём иногда делает доступными свойства, которые недоступны для других методов. В других языках интерфейс, который определяет метод, который Вы хотели бы вызвать, должен быть извлечён (*extracted*) до его использования (см. Лист. 14).

*Лист. 14: В языке Java Вы должны получить интерфейс до того, как Вы его сможете использовать.*

```
XDesktop xDesk;
xDesk = (XDesktop) UnoRuntime.queryInterface(XDesktop.class, desktop);
XFrame xFrame = (XFrame) xDesk.getCurrentFrame();
XDispatchProvider oProvider = (XDispatchProvider)
UnoRuntime.queryInterface(XDispatchProvider.class, xFrame);
```

Если курсор на экране находится в текстовой секции, то свойство *TextSection* курсора содержит ссылку на эту текстовую секцию. В противном случае свойство *TextSection* имеет значение *null*. В языке StarBasic можно получить текстовую секцию следующим способом:

*Лист. 15: OOo Basic позволяет получить прямой доступ к свойствам.*

```
If IsNull(oDoc.CurrentController.getViewCursor().TextSection) Then
```

В других языках (не StarBasic) свойство *CurrentController* и свойство *TextSection* не доступны напрямую. Текущая позиция доступна с использованием метода «*get*», а текстовая секция доступна как значение свойства.

*Лист. 16: Доступ к некоторым свойствам с использованием методов *get*.*

```
oVCurs = oDoc.getCurrentController().getViewCursor()
If IsNull(oVCurs.getPropertyValue("TextSection")) Then
```

Код программы, использующий методы *get* и *set*, легче для переноса в другие языки программирования. StarBasic позволяет также обращаться с некоторыми свойствами как с массивом, даже если данное свойство не является массивом. Хорошим примером является свойство *Sheets* в документе Calc (электронной таблице). Для Calc оба следующих листинга выполняют одинаковые действия, но только второй из них может использоваться без StarBasic.

*Лист. 17: OOo Basic позволяет получить доступ к некоторым свойствам как к массиву.*

```
oDoc.sheets(1)
oDoc.getSheets().getByIndex(1)
```

## I.7. Итоги

Написание макросов для OpenOffice.org (OOo) является сложной задачей, которой непросто научиться. Проблема не в базовом языке или окружении, а в интерфейсе прикладного программирования OOo API. Базовый язык содержит синтаксис и команды, которые не используются для взаимодействия с документом OOo.

*Лист. 18: Простой макрос, который не обращается к OOo API.*

```
Sub SimpleExample()
  Dim i As Integer
  i = 4
  Print "The value of i = " & i
End Sub
```

Большинство макросов написаны для взаимодействия с компонентами OOo и поэтому требуют OOo API. В этом смысле термин API содержит объекты и методы и свойства каждого объекта.

*Лист. 19: Простой макрос, который использует OOo API для просмотра текущего компонента.*

```
Sub ExamineCurrentComponent
  Dim oComp
  oComp = StarDesktop.getCurrentComponent()
  If HasUnoInterfaces(oComp, "com.sun.star.frame.XStorable") Then
    If oComp.hasLocation() Then
      Print "Текущий компонент OOo имеет адрес URL: " & oComp.getLocation()
    Else
      Print "Текущий компонент не имеет адреса местоположения"
    End If
  Else
    Print "Текущий компонент не может быть сохранен"
  End If
End Sub
```

Хотя макрос из Лист. 19 достаточно простой, но требуется знать очень многое, чтобы его написать. Этот макрос начинается с описания переменной oComp, которая по умолчанию имеет тип Variant, потому что явного указания типа нет.



## Глава II. Примеры

### II.1. Отладка и проверка макросов

**М**ожет оказаться трудным определить, какие методы и свойства доступны для данного объекта. В этом разделе описаны способы, которые могут помочь.

#### II.1.1. Определить тип документа

В OOo основная мощь заключена в сервисах. Для определение типа документа достаточно посмотреть, какие сервисы он поддерживает. Приведённый ниже макрос использует этот способ. Это более надёжно, чем использование метода `getImplementationName()`.

*Лист. 20: Определение большинства типов документов OpenOffice.org*

```
'Author: Included with OpenOffice
'Modified by Andrew Pitonyak
Function GetDocumentType(oDoc)
    Dim sImpress$
    Dim sCalc$
    Dim sDraw$
    Dim sBase$
    Dim sMath$
    Dim sWrite$

    sCalc      = "com.sun.star.sheet.SpreadsheetDocument"
    sImpress   = "com.sun.star.presentation.PresentationDocument"
    sDraw      = "com.sun.star.drawing.DrawingDocument"
    sBase      = "com.sun.star.sdb.DatabaseDocument"
    sMath      = "com.sun.star.formula.FormulaProperties"
    sWrite     = "com.sun.star.text.TextDocument"

    On Local Error GoTo NODOCUMENTTYPE
    If oDoc.SupportsService(sCalc) Then
        GetDocumentType() = "sCalc"
    ElseIf oDoc.SupportsService(sWrite) Then
        GetDocumentType() = "sWriter"
    ElseIf oDoc.SupportsService(sDraw) Then
        GetDocumentType() = "sDraw"
    ElseIf oDoc.SupportsService(sMath) Then
        GetDocumentType() = "sMath"
    ElseIf oDoc.SupportsService(sImpress) Then
        GetDocumentType() = "sImpress"
    ElseIf oDoc.SupportsService(sBase) Then
        GetDocumentType() = "sBase"
    End If
    NODOCUMENTTYPE:
    If Err <> 0 Then
        GetDocumentType = ""
        Resume GOON
    GOON:
    End If
End Function
```