

Возможности программирования LibreOffice/OpenOffice для создания программ обработки документов

© В.И. Виноградов, М.В. Виноградова

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Приведен обзор базовых принципов и технологий программирования свободно-распространяемых пакетов LibreOffice и OpenOffice. Рассмотрены возможности пакетов и методы их программирования, в том числе создания макросов и диалогов, добавления внешних библиотек и вызов внешних программ. Даны базовые сведения о технологиях и сервисах универсальных сетевых объектов, а также о событийном управлении. Рассмотренные возможности позволят использовать функции по редактированию документов, преобразованию форматов, импорту и экспорту данных в собственных разработках.

Ключевые слова: *офисный пакет, программирование, кроссплатформенность, обработка документа, макрос, диалог, библиотека, сервис, LibreOffice, OpenOffice, UNO.*

Введение. При разработке программ, предназначенных для формирования и обработки документов, часто возникают задачи их просмотра и редактирования, импорта и экспорта в стандартные форматы. Решение этих задач «с нуля» весьма трудоемко, особенно если речь идет о сложных форматах, поддерживаемых современными текстовыми редакторами.

Альтернативным решением является использование в собственных разработках средств, предоставляемых текстовыми редакторами. Такой подход широко применяется при программировании в среде Microsoft Office на языке Visual Basic for Applications (VBA), который позволяет создавать приложения, использующие возможности Microsoft Access, Word и Excel. Однако Microsoft Office ориентирован на работу в операционной системе Windows и его использование в прочих операционных системах, например, в Linux, затруднительно.

Для создания кроссплатформенных приложений предлагается использовать альтернативу Microsoft Office – офисные пакеты LibreOffice [1] или OpenOffice [2]. Поскольку LibreOffice является ответвлением от проекта OpenOffice и базовый функционал, используемые технологии и интерфейсы программирования у них совпадают, а имеющиеся различия не имеют значения в рамках данной статьи, то оба пакета будут рассмотрены совместно под названием «Офисный пакет Офис» (ОП Офис).

Далее раскрываются возможности и средства ОП Офис, которые могут быть использованы при разработке программ обработки документов.

Возможности ОП Офис. Данный проект является пакетом офисных приложений, объединяющим следующие компоненты [3]: текстовый и табличный процессоры, среду взаимодействия с системой управления базой данных (СУБД), редакторы формул и векторной графики, средство подготовки презентаций. Все компоненты интегрированы в общее ядро и обладают хорошей совместимостью друг с другом. ОП Офис является кроссплатформенным приложением и может быть использован на платформах: Linux (на 32- и 64-разрядных системах), Windows (на 2000 и более поздней версии), MacOS, FreeBSD и Solaris.

Из компонентов ОП Офис для совместного использования с собственными программами обработки документов могут быть задействованы текстовый и табличный процессор [4], а также средство взаимодействия с СУБД [5].

В качестве текстового редактора ОП Офис обладает следующими возможностями:

- создание и редактирование текстовых документов, в том числе содержащих изображения, таблицы, графики и внедренные объекты;
- создание документов слияния;
- использование стандартных шаблонов документов и наличие мастеров по созданию пользовательских шаблонов;
- визуальное редактирование гипертекстовых документов;
- управление объемными и сложными документами, содержащими несколько частей, библиографию, таблицы ссылок и индексы;
- поддержка расчетов и встроенные средства для выполнения вычислений и создания логических связей;
- определение структуры и внешнего вида индексов и таблиц документа;
- управление объектами, вставленными в документ;
- создание и редактирование рисунков в документе и вставка графических объектов, включая изображения в форматах GIF и JPEG;
- проверка орфографии, тезаурус, автозамена и расстановка переносов;
- поддержка стилей форматирования для абзацев, отдельных символов, рамок и страниц;
- навигация по документу, включая закладки и гиперссылки, и работа с документом в режиме структуры;
- обеспечение наборами готовых изображений и текстовых эффектов;
- пользовательская настройка интерфейса, включая состав и размещение панелей инструментов и меню;
- представление удобной справки.

В качестве табличного процессора ОП Офис работает с электронными таблицами, выполняя преобразования и вычисления данных, и может выполнять следующие операции:

- обработка и визуализация табличных данных;
- построение диаграмм, в том числе двух- и трехмерных, включение диаграмм в другие документы;
- поддержка стандартного набора финансовых, статистических и математических функций для вычислений.

ОП Офис обеспечивает инструментальные средства для работы с базами данных (БД) и обладает следующими возможностями:

- создание и редактирование схемы БД и ее компонентов;
- обеспечение пользовательского интерфейса к БД, включая создание и выполнение запросов, форм и отчетов;
- работа с БД под управлением СУБД dBASE, Microsoft Access, MySQL, Oracle;
- работа с БД через драйверы ODBC и JDBC и по технологии ADO;
- использование в качестве источников данных электронных таблиц, текстовых файлов и адресных книг;
- реализация собственной СУБД HSQLDB;
- поддержка стандарта ANSI-92 SQL.

Все компоненты ОП Офис интегрированы между собой, используют общие библиотеки и обеспечивают единую среду, представляющую пользователю объединенный функционал. Подобная интеграция увеличивает функциональность каждого из компонентов ОП Офис по редактированию, преобразованию, импорту и экспорту документов различных форматов.

Рассмотренные выше компоненты ОП Офис поддерживают работу с документами следующих форматов [6]:

- текстовый документ ODF (.odt);
- шаблон текстового документа ODF (.ott);
- электронная таблица ODF (.ods);
- шаблон электронной таблицы ODF (.ots);
- текстовый документ OpenOffice.org 1.0 (.sxw);
- шаблон текстового документа OpenOffice.org 1.0 (.stw);
- электронная таблица OpenOffice.org 1.0 (.sxc);
- шаблон электронной таблицы OpenOffice.org 1.0 (.stc);
- база данных ODF (.odb);
- документы Microsoft Word (.doc и .docx);
- Microsoft Excel (.xls и .xlsx);
- шаблон Microsoft Excel (.xlt и .xltx);
- документы Microsoft Word XML и Excel XML (.xml);
- HTML документ (.html и .xhtml);

- Rich Text Format (.rtf);
- текст (.txt);
- Data Interchange Format (.dif);
- dBASE (.dbf);
- SYLK (.slk);
- текст csv (.csv);

Кроме того, поддерживается возможность экспорта документов в PDF.

Для обеспечения дополнительной функциональности ОП Офис предоставляет возможность разрабатывать независимые сторонние расширения, которые легко устанавливаются и удаляются, а также встроенные средства для наращивания функционала через программирование ОП Офис.

Возможности программирования ОП Офис. ОП Офис поддерживает следующие возможности расширения своего функционала [7]:

- создание макросов;
- создание диалогов;
- добавление внешних библиотек;
- вызов внешних программ.

Макросы – программные сценарии, написанные на одном из четырех языков: Basic, Python, BeanShell или JavaScript. Наиболее используемым из них является Basic, синтаксис которого похож на синтаксис языков Microsoft VBA и Microsoft VB. Данный язык является процедурным, но позволяет использовать интерфейс прикладного программирования (API) объектных библиотек. Его конструкции позволяют объявлять переменные, в том числе объекты, организовывать циклы и условия, объявлять и вызывать процедуры и функции, обрабатывать ошибки, работать с объектами, выполнять логические и арифметические операторы. Разработчику предоставляется большой набор системных функций, в частности, преобразование данных, работа с файлами и обращение к внешней среде [8].

Для создания и отладки макросов используется интегрированная среда разработки [6], которая является частью ОП Офис.

При выполнении макросов в среде ОП Офис они построчно интерпретируются и их команды исполняются стандартными компонентами ОП Офис [8]:

- библиотека времени выполнения, реализующая стандартные функции;
- API ОП Офис, который обеспечивает прямой доступ к документам и компонентам ОП Офис по технологии универсальных сетевых объектов (Universal Network Objects, UNO).

Макросы позволяют программно осуществлять все те действия, которые можно выполнить через стандартный пользовательский ин-

терфейс ОП Офис, в том числе редактирование документов, управление окнами, перехват событий и отмена выполнения стандартных команд [9].

Макросы управляют работой ОП Офис через объектную модель по технологии UNO. Технология UNO определяет объектно-ориентированный интерфейс для программно-управляемого доступа к ОП Офис. Все элементы, компоненты и документы ОП Офис (редактируемые документы и их части, окна и панели редактирования, модули ядра и подключенные библиотеки) доступны для управления и представлены набором связанных объектов, имеющих методы и свойства [10].

Для обращения к любому объекту ОП Офис необходимо создать соответствующий сервис. Сервисы создаются через вызов специальной функции, которая в качестве входного параметра принимает название сервиса и возвращает указатель на созданный объект сервиса. Через последующий вызов методов сервиса можно загружать объекты компонентов ОП Офис и далее обращаться к их методам и свойствам, в том числе являющихся вложенными объектами. Примерами объектов ОП Офис могут служить:

- объект главного окна, являющийся объектом верхнего уровня и управляющий загрузкой документов и отображением окон;
- объекты загружаемых документов, позволяющие сохранять, экспортировать и печатать документы;
- объекты соединения с БД и объекты текстовых символов, абзацев и ячеек электронных таблиц [11].

Некоторые объекты доступны без явной загрузки сервисов, например, объекты главного окна и текущего документа, являющиеся глобальными переменными. Некоторые сервисы могут быть загружены неявно, если они являются зависимыми, и доступны через сетевые объекты, предоставляющие сервисы более высоких уровней.

Базовые понятия в технологии UNO – интерфейс, сервис и модуль. Интерфейс определяет набор доступных методов и их сигнатуры. Именно эти методы вызываются из макросов. Каждый сервис реализует один или более интерфейсов и содержит свойства, используемые при обращении к сервису: структуры, исключения, константы и перечисления. Сервисы иерархически объединены в модули, вложенные друг в друга в соответствии с иерархией. Вершиной иерархии является центральный модуль, в который вложены все прочие модули. Структура иерархии задается правилами именования сервисов, сходные с правилами именования иерархии пакетов в языке Java, например, «сервис_главный.сервис2уровня.сервис3уровня.сервис4уровня» [12].

При работе с объектами ОП Офис можно перейти от объектов верхнего уровня иерархии к вложенным объектам, используя интерфейсы первых. Так, для конкретного документа можно работать с

коллекциями его абзацев, рисунков или таблиц. Для обращения к вложенным объектам используются их имена или индексы, поддерживаются методы добавления, удаления, обхода и поиска по коллекциям вложенных элементов.

Для интеграции с собственными программами обработки документов могут быть использованы следующие сервисы:

- редактирования текста документа;
- форматирования и настройки свойств символа, абзаца, ячейки таблицы, страницы;
- настройки свойств печати;
- навигации и поиска в пределах документа и его частей;
- работы с текстовыми полями (как предопределенными, так и пользовательскими);
- работы с таблицами и их элементами вплоть до отдельных ячеек;
- создания диаграмм;
- создания и вставки графических объектов;
- работы с базой данных;
- управления элементами форм и диалогов.

Сервисы управления интерфейсом пользователя предоставляют возможности настраивать внешний вид приложения ОП Офис, включая отображение или сокрытие пунктов меню, строки статуса и панелей инструментов, создавать собственные пункты меню и панели инструментов.

В ОП Офис реализована событийная модель управления, по которой определяется набор возможных событий и для каждого из них может быть назначен обработчик. Выделяют события, связанные как с действиями пользователя (например, нажатие клавиши или движение мыши), так и с обработкой данных (например, открытие или сохранение документа).

Для события можно определять перехватчики и обработчики [13]. Те и другие являются процедурами и вызываются при возникновении указанного события. Отличие перехватчиков от обработчиков заключается в том, что первые добавляются в цепочку процедур обработки события наряду со множеством прочих, в том числе стандартных, а вторые могут прервать цепочку вызовов, вернув признак отказа от дальнейшей обработки. Таким образом, обработчики могут прервать стандартное выполнение команды. И обработчики, и перехватчики событий назначаются через установку требуемых свойств объектов соответствующих сервисов.

Установку некоторых обработчиков событий, в том числе нажатия «горячих» клавиш, запуска и закрытия приложения, действий с файлами документов, а также создание пользовательских меню и панелей можно выполнять через стандартные средства настройки ОП

Офис. Все прочие события перехватываются через программирование объектов сервисов.

Кроме обработки документов макросы используются для организации и управления диалогами.

Диалоги – экранные формы, содержащие элементы графического пользовательского интерфейса. Диалоги могут содержать поля ввода разных форматов, списки и поля со списками, надписи и изображения, таблицы данных, кнопки и поля выбора файла, флажки и переключатели, полосы прокрутки и т.д. ОП Офис имеет встроенный редактор диалогов, позволяющий создавать и редактировать пользовательские диалоговые окна, добавлять элементы управления и обработчики событий. Работа в редакторе диалогов ОП Офис сходна по возможностям и технологии применения с программированием в визуальной среде разработки, например, Qt Creator или Microsoft Visual Studio.

Диалог и каждый его элемент имеют набор predetermined событий от мыши и клавиатуры или изменения фокуса, а также специфических – для конкретного элемента управления, например, выбор элемента в списке. Для каждого из перечисленных событий средствами редактора диалогов можно назначить обработчика. Обработчики событий описываются как процедуры, принимающие на вход параметр – сетевой объект, предоставляющий сервис для получения детальной информации о возникшем событии и его источнике, например, код нажатой клавиши и свойства текущего элемента управления.

Создание диалогов и управление их свойствами, в том числе храняемыми в них данными возможно через программирование сервисов по технологии рассмотренной выше UNO. Сервисы управления диалогами позволяют загружать и выполнять диалоги, обращаться к их элементам, запускать из диалогов макросы и использовать в диалогах результаты выполнения макросов, настраивать встроенные диалоги ОП Офис. Для обращения к содержимому диалога и его элементов используется свойство «Модель», присущее всем элементам управления, а для настройки их внешнего вида – их свойства «Вид».

Поскольку макросы и диалоги тесно связаны, то далее они будут рассматриваться совместно и именоваться макросами.

Макросы можно привязать к конкретному документу. Тогда они сохраняются вместе с документом и доступны при его открытии в любом экземпляре ОП Офис. Альтернативой является привязка макросов к конкретному приложению ОП Офис. В этом случае макросы доступны при работе с любыми документами, обрабатываемыми данным экземпляром ОП Офис.

Код макросов и описания диалогов хранятся в файлах, называемых модулями. Модули могут быть помещены в библиотеку – каталог, где кроме файлов модулей хранятся файлы с описанием исполь-

зуемых макросов и диалогов. Библиотеки можно распространить среди нескольких пользователей по сети, в том числе в автоматическом режиме через внесение изменений в конфигурационные файлы ОП Офис. Для этого необходимо загрузить файлы библиотек в определенный каталог ОП Офис на удаленной рабочей станции и добавить ссылки на них в конфигурационные файлы на той же станции, хранящиеся в формате XML. Библиотеки позволяют объединять макросы и диалоги на основе функциональной связанности и централизованно подключать их к пользовательским экземплярам ОП Офис.

Макросы в ОП Офис могут выполняться в стандартном, скрытом или фоновом режимах. При работе в *стандартном режиме* отображаются диалоговые окна, содержащие окна редактора, панели инструментов и меню. При работе в *скрытом режиме* происходит выполнение команд макроса без видимого отображения. Для запуска макроса в *фоновом режиме* необходимо средствами операционной системы сформировать задание и расписание его запуска.

Внешние библиотеки, подключаемые к ОП Офис, могут быть написаны на языках C++ или Java [14]. Они импортируются в ОП Офис и далее используются наравне со стандартными компонентами. Для подключения библиотек необходимы файлы конфигурации, которые определяют сигнатуры интерфейсов на языке UNOIDL, регистрацию схемы протокола и шаги инсталляции. Подключаемые библиотеки кроме собственных функций должны реализовывать набор базовых интерфейсов UNO для обеспечения взаимодействия с ОП Офис [15]. Схема протокола задает правила использования и вызова сервисов и соответствующих им исполняемых компонентов. Эти правила используются диспетчером событий ОП Офис для обращения к сервисам компонентов. Для написания и компиляции программ библиотек, описания интерфейсов и подготовки файлов конфигурации используют средства Среды разработки (SDK) ОП Офис.

Вызов внешних программ выполняется из макросов ОП Офис в режиме командной строки. При этом возможна передача параметров. Внешнее приложение должно являться утилитой и поддерживать разбор командной строки.

Внешние программы могут использовать средства ОП Офис такими способами, как:

- запуск макросов ОП Офис из командной строки в обычном или скрытом режимах;
- обращение к сервисам ОП Офис по технологии UNO.

Для запуска макроса из командной строки внешнее приложение должно запустить процесс ОП Офис и в качестве параметра передать ему название макроса и (при необходимости) аргументы макроса. Запуск процесса и передача ему параметров выполняется стандартными средствами среды разработки приложения.

Обращение к сервисам ОП Офис по технологии UNO возможно из приложений, написанных на языках C++, Java и Basic [16, 17]. Кроме того, можно программировать с использованием .NET языков через CLI и на сценарных языках JavaScript, Beanshell и Python.

Для обращения к объектам ОП Офис необходимо [18]:

- настроить среду окружения (например, установить значения переменных среды окружения) для использования библиотек ОП Офис;
- создать заголовочные файлы, содержащие описания интерфейсов API ОП Офис на используемом языке программирования;
- создать заголовочные файлы, содержащие описания типов ОП Офис;
- запустить сервер ОП Офис и настроить его порт для прослушивания запросов;
- создать новое приложение в среде разработки, подключить к нему исполняемые библиотеки ОП Офис и созданные заголовочные файлы, настроить конфигурацию среды разработки;
- добавить в приложение программный код для работы с сервисами ОП Офис, используя сигнатуры сервисов и типы данных, описанные в заголовочных файлах.

Заголовочные файлы создаются средствами Среды разработки (SDK) ОП Офис, которая необходима для программирования ОП Офис. Принципы работы с сервисами аналогичны тем, что были рассмотрены для создания макросов.

Особенности реализации каждого из описанных шагов зависят от конкретного языка программирования.

Включение функционала ОП Офис в собственные разработки. Подводя итог рассмотрению возможностей использования пакетов LibreOffice или OpenOffice при разработке собственного приложения, работающего с документами стандартных форматов (в том числе doc, pdf, odt, xls и html), можно предложить следующие варианты их применения:

- подключение функционала и элементов графической среды офисных пакетов для просмотра, редактирования и печати документов;
- формирование сложных составных документов на основе содержимого БД;
- взаимодействие с пользовательской базой данных для импорта, экспорта и обработки документов;
- преобразование форматов документов;
- разбор документов и выполнение слияния с файлами данных по стандартным или пользовательским алгоритмам.

Включение функционала офисного пакета в собственное приложение возможно следующими методами:

1) написание макроса (в среде офисного пакета), запускаемый основным приложением из командной строки. Макрос подгружает необходимые ему библиотеки и обращается к их сервисам;

2) создание функций в среде основного приложения (на языке C++ или Java). Функции работают с сервисами через проекцию классов сетевых объектов.

Рассмотренные варианты отличаются средой разработки и используемым языком программирования. Алгоритмы функций и макросов, а также набор используемых в них сервисов совпадают. Поскольку разработка и выполнение макросов в родной среде представляются менее трудоемкими и более надежными, то на начальной стадии разработки рекомендуется использовать первый метод.

Рассмотренные технологии программирования пакетов LibreOffice и OpenOffice позволяют интегрировать их функции и компоненты в собственные проекты для повышения эффективности процесса разработки прикладных программ и их последующей эксплуатации.

ЛИТЕРАТУРА

- [1] Официальный сайт проекта LibreOffice. URL: http://www.u.wikipedia.org/wiki/Libre_Office (дата обращения 30.06.2012).
- [2] Материал из Википедии по OpenOffice.org. URL: http://ru.wikipedia.org/wiki/Libre_Office (дата обращения 20.06.2012).
- [3] Ионов А., Коновалов Ю., Новодворский А., Трунин И., Смирнов Д., Якушин А., Гарбуз Н. *Практическое руководство по Ru.OpenOffice.org*. Москва, Инфра-Ресурс, 2005, 84 с.
- [4] Справочное руководство по LibreOffice. URL: https://help.libreoffice.org/Main_Page/ru (дата обращения 20.07.2012).
- [5] Пьяных Е.Г. *Проектирование баз данных в среде OpenOffice.org* (ПО для управления базами данных): учеб. пособие. Москва, 2008, 62 с.
- [6] *OpenOffice.org. Начинаем работу*. Пер. с англ. Москва, Инфра-Ресурс, 2007, 331 с.
- [7] Бейн М.А. Изучение программирования макросов для электронных таблиц в OpenOffice.org. OOo Basic и Автоматизация Calc. Быстрое и дружественное руководство по написанию макросов и приложений электронных таблиц. Пер. с англ. Санкт-Петербург, Packt Publishing, 2008, 151 с.
- [8] Программирование на языке OpenOffice.org Basic. Пер. с англ. Москва, Инфра-Ресурс, 2007. 184 с.
- [9] Питоньяк Э. *OpenOffice.org pro. Автоматизация работы*. Москва, ДМК Пресс, 2008, 512 с.
- [10] Справочная информация по API OpenOffice.org. URL: http://wiki.openoffice.org/wiki/Main_Page (дата обращения 30.06.2012).
- [11] Справочная информация по модулям Apache OpenOffice. URL: <http://www.openoffice.org/api/docs/common/ref/com/sun/star/module-ix.html> (дата обращения 06.07.2012).
- [12] Apache OpenOffice Developer's Guide. URL: http://wiki.openoffice.org/wiki/Documentation/DevGuide/OpenOffice.org_Developers_Guide (дата обращения 3.07.2012).

- [13] Питоньяк Э. *Полезная информация по макросам для OpenOffice*. Пер. с англ. URL: http://www.pitonyak.org/AndrewMacro_rus.odt (дата обращения 22.07.2012).
- [14] Начала программирования в OpenOffice.org. Пер. с англ. М., Инфра-Ресурс, 2007, 37 с.
- [15] Справочная информация по созданию шлюзов через UNO. URL: http://wiki.openoffice.org/wiki/Uno/Article/About_Bridges (дата обращения 05.07.2012).
- [16] Справочная информация по программированию на языке C++ по технологии UNO. URL: <http://www.openoffice.org/api/docs/cpp/ref/names/index.html> (дата обращения 05.07.2012).
- [17] Справочная информация по программированию на языке Java по технологии UNO. URL: <http://www.openoffice.org/api/docs/java/ref/overview-summary.html> (дата обращения 05.07.2012).
- [18] Автоматизация OpenOffice. URL: <http://habrahabr.ru/post/116151/> (дата обращения 04.07.2012).

Статья поступила в редакцию 18.12.2013

Ссылку на эту статью просим оформлять следующим образом:

Виноградова М.В., Виноградов В.И. Возможности программирования LibreOffice/OpenOffice для создания программ обработки документов. *Инженерный журнал: наука и инновации*, 2013, вып. URL: <http://engjournal.ru/catalog/it/hidden/1060.html>

Виноградова Мария Валерьевна родилась в 1976 г., окончила факультет информатики и систем управления МГТУ им. Н.Э. Баумана в 2000 г. Канд. техн. наук, доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана. Автор более 20 научных и учебно-методических работ в области проектирования информационных систем и постреляционных баз данных. e-mail: vinogradova.m@bmstu.ru

Виноградов Валерий Иванович, родился в 1941 г., окончил факультет приборостроения МВТУ им. Н.Э. Баумана в 1965 г. Канд. техн. наук, доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана. Автор более 50 научных и учебно-методических работ по направлениям Архитектура ЭВМ и разработка автоматизированных систем. e-mail: vinogradovs.fam@bmstu.ru