

**Технология использования компонентов и функций пакетов  
LibreOffice/OpenOffice при разработке программ обработки документов  
77-48211/637941**

**# 08, август 2013**

**Виноградова М. В., Виноградов В. И.**

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

[vinogradova.m@bmstu.ru](mailto:vinogradova.m@bmstu.ru)

[vinogradovs.fam@bmstu.ru](mailto:vinogradovs.fam@bmstu.ru)

### **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 этими самими 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 «Программный пакет Офис» (ПП Офис).

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

### **1 Возможности ПП Офис**

ПП Офис является пакетом офисных приложений, объединяющим следующие компоненты [3]: текстовый и табличный процессоры, среду взаимодействия с СУБД, редакторы формул и векторной графики, средство подготовки презентаций. Все

компоненты интегрированы в общее ядро и обладают хорошей совместимостью друг с другом. ПП Офис является кроссплатформенным приложением и может быть использован на платформах: Linux (на 32- и 64-разрядных системах), Windows (2000 и более поздняя), MacOS, FreeBSD и Solaris.

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

Для обеспечения дополнительной функциональности ПП Офис предоставляет возможность разрабатывать независимые сторонние расширения, которые легко устанавливаются и удаляются, а также встроенные средства для наращивания функционала через программирование ПП Офис [6].

ПП Офис поддерживает следующие возможности расширения его функционала [7, 8, 9, 10]:

- Создание макросов;
- Создание диалогов;
- Добавление внешних библиотек;
- Вызов внешних программ.

Все компоненты ПП Офис интегрированы между собой, используют общие библиотеки и обеспечивают единую среду, представляющую пользователю объединенный функционал. Подобная интеграция увеличивает функциональность каждого из компонентов ПП Офис по редактированию, преобразованию, импорту и экспорту документов различных форматов [11].

## **2 Класс программ для создания отчетов**

Класс прикладных программ, выполняющих формирование, просмотр и печать отчетов на основе данных из таблиц БД, достаточно широк. Спектр подобных программ включает в себя как маленькие программы, используемые в небольших учреждениях или отделах и состоящие из набора экранных форм для просмотра и редактирования данных, так и большие программные комплексы, обеспечивающие документооборот всей организации [12]. Однако, независимо от количества экранных форм, объема используемых данных и сложности их преобразования все эти программы выполняют общий набор функций, что позволяет объединить их в один класс.

Программы рассматриваемого класса далее будут называться программами создания отчетов (ПСО). Это могут быть либо независимые программы и программные

комплексы, либо подсистемы и модули, используемые совместно с другими программными компонентами в рамках более крупных систем.

В общем случае ПСО должна обеспечивать выполнения следующих функций:

- Формирование и выполнение SQL запросов произвольной структуры для получения табличных данных из БД;
- Сохранение результатов запросов в файлы стандартных форматов, например, xml и rtf;
- Формирование отчетов на основе полученных файлов в виде документов стандартных форматов, например, doc или html;
- просмотр и печать полученных отчетов.

Полученные отчеты должны быть совместимы со стандартными редакторами для последующего просмотра и редактирования в этих редакторах.

На основе функций ПСО предлагается выделить следующие логические компоненты:

- набор функций для реализации функционала - API ПСО;
- пользовательский интерфейс для взаимодействия с оператором - ПИ ПСО;
- раздел БД ПСО для хранения мета-информации о запросах, шаблонах и форматах отчетов;
- редактор для просмотра и печати отчетов.

Взаимодействие компонентов ПСО между собой показано на рисунке 1.

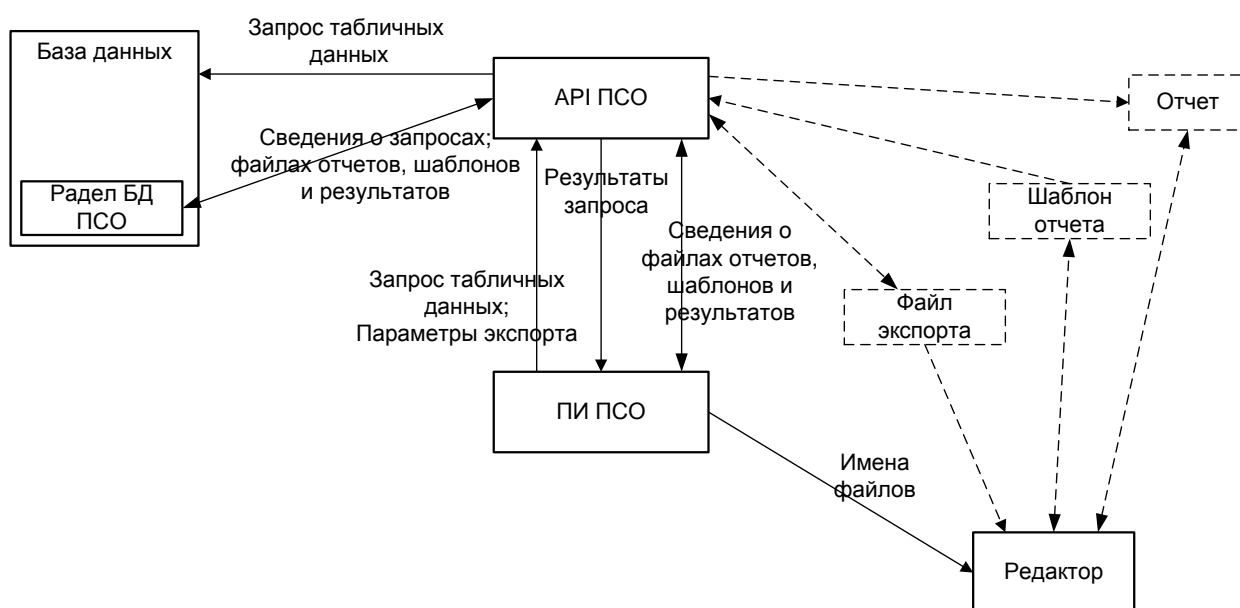


Рисунок 1. Схема взаимодействия компонентов ПСО

ПИ реализует интерфейс с пользователем для ввода и интерпретации команд пользователя и отображения результатов их выполнения. Он обращается к API и Редактору.

ПИ передает API текст и параметры SQL-запроса к базе данных и параметры экспорта данных (формат экспорта, путь к файлу экспорта и настройки экспорта, зависящие от конкретного формата). Также ПИ обращается к разделу БД ПСО через API для записи и получения сведений о форматах и шаблонах отчетов.

API обращается к различным разделам базы данных организации для выполнения пользовательских SQL-запросов и получения результатов. Полученные результаты возвращаются в ПИ или сохраняются в файле экспорта стандартного формата. Также API осуществляет поиск сведений в разделе БД ПСО.

ПИ вызывает функции редактора для формирования шаблонов слияния с файлами экспорта при формировании отчетов, просмотра и печати отчетов. Функции редактора могут быть выполнены либо стандартным редактором, либо ПСО. В последнем случае API выполняет слияние шаблона отчета и файла экспорта для создания отчета. Формирование шаблонов, просмотр и печать отчетов и файлов экспорта в этом случае выполняются ПИ. Возможно делегирование части функций стандартному редактору.

### **3 Технология взаимодействия ПСО с ПП Офис**

#### **3.1 Выполнение SQL-запросов к БД при экспорте**

Рассмотрим перечень задач, выполнение которых можно передать ПП Офис при разработке прикладной программы по созданию и обработке документов. Первой из них является выполнение запросов к БД комплекса для создания файлов экспорта и отчетов. Возможны два варианта:

Вариант 1: через ПИ и API из раздела БД ПСО извлекаются данные о запросах, которые передаются в ПП Офис. ПП Офис запрашивает параметры выполнения запросов, выполняет SQL-запросы, на основе полученных результатов создает файлы отчетов и/или экспорта. Схема взаимодействия компонентов представлена на рисунке 2.

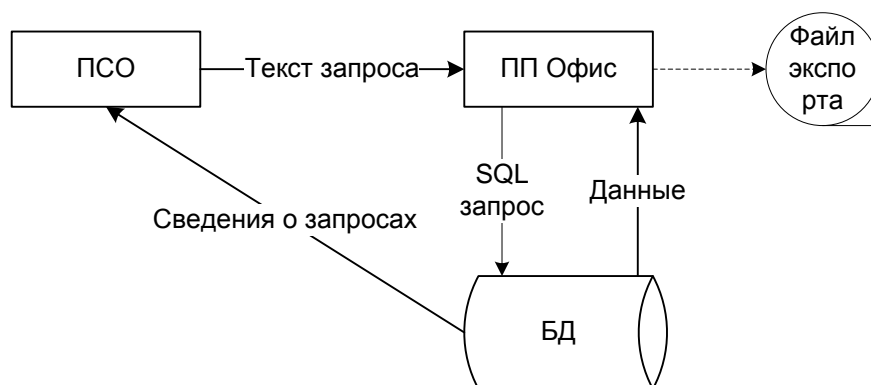


Рисунок 2. Выполнение запросов по варианту 1

Преимущества: разделение функций по ведению раздела БД ПСО и работы с документами между ПСО и ПП Офис.

Недостатки: дублирование функций работы с БД; децентрализация процесса обеспечения безопасности при обращении к БД; необходимость программирования на стороне ПП Офис бизнес-логики и пользовательского интерфейса; «утолщение» функционала ПП Офис.

– Вариант 2: ПИ через API выполняет SQL-запросы к БД комплекса. Результаты выполнения запроса передаются в ПП Офис. ПП Офис на основе полученных данных создает файлы отчетов и/или экспорта. Схема взаимодействия компонентов представлена на рисунке 3.

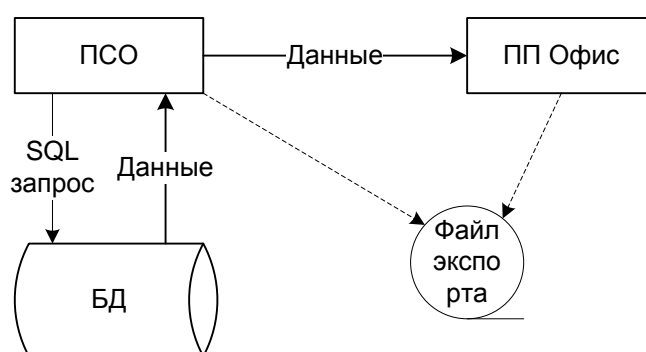


Рисунок 3. Выполнение запросов по варианту 2

Преимущества: централизация функций работы с БД и пользовательского интерфейса; возможность создания файлов экспорта и слияния документов средствами ПСО; сокращение функций, выполняемых средствами ПП Офис до минимума (только некоторые преобразования форматов и слияния документов).

Недостатки: Распыление функций экспорта и редактирования между ПСО и ПП Офис.

По результатам сравнения выбирается второй вариант – выполнение SQL-запросов на стороне ПСО. Этот вариант позволит избежать дублирования функций и обеспечит единую точку доступа к БД.

Данные – результаты выполнения SQL-запросов, между ПСО и ПП Офис предлагается передавать в формате XML [13]. Формат XML позволяет передавать как табличные данные, так и сведения об их структуре и типах, а также легко расширяем, прост в обработке и независим от среды разработки. Возможно использование встроенных средств работы с XML-документами как в модулях ПСО, так и в макросах ПП Офис.

### **3.2 Экспорт результатов выполнения запросов в файлы**

Средствами ПСО необходимо выполнять экспорт результатов выполнения запросов в файлы стандартных форматов, например, RTF [14], XML, CSV [15]. Форматы XML и CSV являются текстовыми, достаточно просты и могут обрабатываться функциями среды разработки. По этой причине экспорт в файлы форматов XML и CSV предлагается выполнять средствами API ПСО.

Формат RTF более сложен для обработки, поэтому предлагается работать с ним, используя сервисы ПП Офис.

Предлагается следующее распределение обязанностей между компонентами ПСО при экспорте табличных данных:

ПИ ПСО позволяет пользователю ввести текст и параметры SQL-запроса (или выбрать их из хранящихся в разделе БД ПСО); выбрать формат для экспорта и имя файла для сохранения результата; указать параметры экспорта (зависящие от конкретного формата);

API ПСО выполняет SQL-запрос и возвращает его результат в объекте формата XML;

- Если выбран формат RTF, то XML-данные, имя файла и параметры экспорта передаются в ПП Офис, который создает новый документ в формате RTF, выводит в него табличные данные и сохраняет на диске;

- Если выбран формат CSV или XML, то данные, имя файла и параметры экспорта передаются в API ПСО, который создает новый файл, выводит в него табличные данные в соответствии с указанными параметрами экспорта и сохраняет файл на диске.

Если перед сохранением в файл необходимо отобразить результаты выполнения SQL-запроса в табличной форме, то это делает ПИ ПСО на основе сведений о структуре данных, содержащихся в XML-объекте.

### **3.3 Импорт результатов выполнения запросов**

Под импортом результатов выполнения запросов понимается открытие файлов экспорта и просмотр их содержимого в текстовой или табличной форме. Поскольку файлы экспорта сохраняются на локальном ПК пользователя, то открыть их для просмотра в текстовом виде можно в любом текстовом редакторе вне ПСО, например, ПП Офис.

Файлы в форматах CSV или XML в текстовой форме можно также просматривать в окне текстового редактора ПИ ПСО. Файлы в формате RTF предлагается перенаправлять для просмотра в окно редактора ПП Офис.

Для отображения содержимого файлов экспорта в табличной форме средствами ПСО необходимо предварительно выполнять преобразование в объект формата XML. Преобразование из форматов CSV или XML выполняется функциями API ПСО, преобразование из формата RTF –сервисами ПП Офис.

Перед выполнением импорта ПИ ПСО позволяет пользователю выбрать файл экспорта, указать используемые параметры экспорта и выбрать форму отображения.

### **3.4 Редактирование файлов шаблонов**

Средствами ПСО необходимо создавать файлы шаблонов отчетов в форматах XSL [16] и ODF [17]. Под созданием понимается создание нового или открытие существующего файла, редактирование его содержимого и сохранения на диске.

Формат XSL является текстовым и достаточно простым, поэтому для его создания и редактирования можно использовать средства текстового редактора ПСО. Формат шаблонов ODF богаче по возможностям и более сложен, поэтому для работы с ним предлагается использовать сервисы ПП Офис.

Для поддержания целостности данных и согласованности раздела БД ПСО обращение к файлам шаблонов следует выполнять только через ПИ ПСО. Для редактирования файла шаблона необходимо предлагается следующее распределение функций:

ПИ ПСО позволяет пользователю выбрать шаблон для редактирования (или создать новый). На основе идентификатора файла для выбранного шаблона выполняется обращение к БД ПСО для извлечения файла шаблона в локальный файл. Извлеченный файл открывается в окне редактирования для просмотра и внесения изменений. После сохранения внесенных изменений в файл и закрытия окна редактора файл шаблона сохраняется в БД ПСО. Для внесения изменений в раздел БД ПСО вызывается API ПСО.

API ПСО выполняет чтение и запись в раздел БД ПСО сведений о шаблоне и идентификаторе его файла;

- Текстовый редактор ПСО обеспечивает работу с текстом в формате XSL;
- Сервисы ПП Офис обеспечивают редактирование файла в формате ODF.
- 

### **3.5 Формирование отчетов**

Формирование отчетов происходит методом слияния шаблона отчета и результатов выполнения запросов с последующим сохранением полученного документа в формате ODF [17] или HTML[18].

Перед выполнением слияния необходимо подготовить средствами ПИ ПСО исходные данные для построения отчета: извлечь файл шаблона, ввести тексты и параметры запросов. Функция Создания отчетов API ПСО выполняет запросы; сохраняет их результаты в одном или более объектах формата XML; открывает файл шаблона; выполняет слияние шаблона с результатами и сохраняет созданный документ в файл. Файл отчета является локальным и сохраняется в файловой системе локального ПК.

Возможно два варианта слияния:

- Слияние с шаблоном в формате XSL. Результат слияния сохраняется в формате HTML. Работа с файлами и слияние выполняется API ПСО;
- Слияние с шаблоном в формате ODF. Результат слияния сохраняется в формате ODF. Для выполнения слияния используются сервисы ПП Офис. Сервисы обеспечивают открытие файла шаблона; слияние с документами в формате XML, сохранение полученного отчета в файл формата ODF.
- 

### **3.6 Просмотр, редактирование и печать отчетов**

Для просмотра отчетов в формате HTML предлагается использовать стандартный Интернет-браузер, который также предоставляет возможность настройки и выполнения печати. Открывать отчеты в браузере можно как вне ПСО, так и из ПИ ПСО. В последнем случае ПИ ПСО вызывает приложение Интернет-браузер и передает ему в качестве параметра файл отчета, сохраненный на диске.

Для просмотра и печати отчетов в формате ODF предлагается использовать средства стандартных текстовых редакторов, например, ПП Офис. Поскольку отчеты сохраняются в локальных файлах, то работать с ними можно вне ПСО. При необходимости просмотра и печати отчета средствами ПСО последний вызывает текстовый редактор ПП Офис и передает ему в качестве параметра файл отчета.

Нет необходимости контролировать доступ к отчетам средствами ПСО.



#### **4 Настройка приложения ПП Офис**

ПП Офис используется в ПСО для выполнения:

- Экспорта и преобразования данных;
- Слияния с шаблоном;
- Просмотра, редактирования и печати документов.

Экспорт и преобразование данных и слияние с шаблоном можно выполнять программными модулями, которые используют сервисы ПП Офис и взаимодействуют с ними по технологии UNO. Подключение пользовательского интерфейса ПП Офис не требуется. Возможно два варианта:

- Программные модули являются макросами ПП Офис, например, написанными на языке Basic, и работают в скрытом режиме. Запуск макроса выполняется из командной строки. Данные для обработки передаются через локальные файлы. Для запуска макроса создается и запускается отдельный процесс. Макрос подгружает необходимые библиотеки и обращается к их сервисам;

Преимущества: написание и выполнение макроса происходит в его родной среде, что повышает надежность и упрощает программирование и отладку; макрос независим от ПСО и является самостоятельным приложением; передача данных через файлы уменьшает сцепление макросов и ПСО по структурам данных.

Недостатки: необходимо запускать отдельный процесс для вызова макроса; требуется обращение к файловой системе для передачи параметров; разработка ведется в двух разных средах разработки.

- Программные модули реализованы как библиотека функций, написанная на языке C++ (или Java) [19, 20, 21, 22]. Функции работают с сервисами ПП Офис через проекцию классов сетевых объектов. Обращение к библиотеке функций происходит через ее программный интерфейс. Данные для обработки могут передаваться как объекты или как локальные файлы.

Преимущества: все приложение написано в одной среде; объекты ПП Офис встроены в ПСО; возможна передача параметров без обращения к файловой системе; нет вызова внешних процессов.

Недостатки: процесс программирования усложняется за счет применения более сложной технологии и необходимости настройки среды разработки на работу с сетевыми объектами; при передаче параметров как объектов усиливается сцепление между ПСО и функциями библиотеки; при вызове сервисов сетевых объектов удлиняется цепочка обработки за счет работы с проекциями.

С точки зрения программирования модулей рассмотренные варианты отличаются средой разработки и используемым языком программирования. Алгоритмы функций и макросов и набор используемых в них сервисов совпадают. С точки зрения использования различия заключаются в различных технологиях вызова (через командную строку или API) и способе передачи параметров.

Поскольку разработка и выполнение макросов в родной среде представляются менее трудоемкой и более надежной соответственно, то на начальной итерации разработки ПСО предлагается использовать первый вариант (написание и вызов макросов). При необходимости внедрить объекты ПП Офис в ПСО и обращаться к ним через API перевод макросов на другой язык будет достаточно простым из-за однозначности алгоритмов.

### **Заключение**

На основе анализа класса программ по обработке документов и формированию отчетов были определены их типовые функции, включая выполнение SQL-запросов к БД, отображению и экспорту результатов, формированию отчетов стандартных форматов. На основе рассмотрения технологий программирования пакетов LibreOffice и OpenOffice были определены возможности по использованию их функций и компонентов в собственных разработках.

Разработаны методы и технология интеграции стандартного редактора с прикладной программой для выполнения типовых функций создания отчетов. Проведен сравнительный анализ вариантов интеграции и приведено обоснование предлагаемых решений.

Рассмотренные методы интеграции пакетов LibreOffice и OpenOffice с прикладными программами создания отчетов позволяют сократить время и стоимость разработки программ за счет сокращения затрат на проектирование и программирование включаемых компонентов. А также позволяют повысить эффективность их эксплуатации за счет использования знакомых пользователям экранов и стандартных кроссплатформенных средств, поддерживаемых официальными разработчиками пакетов.

### **Список литературы**

1. Материалы официального сайта проекта LibreOffice. Режим доступа: <http://www.libreoffice.org> (Дата обращения 30.10.2013).
2. Материалы официального сайта проекта OpenOffice.org. Режим доступа: <http://openoffice.org> (Дата обращения 20.06.2013).

3. Ионов А., Коновалов Ю., Новодворский А., Трунин И., Смирнов Д., Якушин А., Гарбуз Н. Практическое руководство по Ru.OpenOffice.org, М.: Инфра-Ресурс, 2005. – 84 с.
4. Бейн Марк А. Изучение программирования макросов для электронных таблиц в OpenOffice.org. OOo Basic и Автоматизация Calc. Быстрое и дружелюбное руководство по написанию макросов и приложений электронных таблиц: пер. с англ, СПб.: Санкт-Петербург, 2008 г. 151 с. [Dr. Mark Alexander Bain Learn OpenOffice.org. Spreadsheet Macro Programming. OOoBasic and Calc Automation. A fast and friendly tutorial to writing macros and spreadsheet applications. - BIRMINGHAM — MUMBAI: Packt Publishing, 2006 ]
5. Справочное руководство по LibreOffice. Режим доступа: [https://help.libreoffice.org/Main\\_Page/ru](https://help.libreoffice.org/Main_Page/ru). (Дата обращения 20.07.2013).
6. Программирование на языке OpenOffice.org Basic: пер. с англ. - М.: Инфра-Ресурс, 2007. – 184 с. [ StarOffice 8 Programming Guide for BASIC. - Sun Microsystems]
7. Питоньяк Э. OpenOffice.org pro. Автоматизация работы. - ДМК Пресс, 2008. - 512 с.
8. Справочная информация по API OpenOffice.org. Режим доступа: [http://wiki.openoffice.org/wiki/Main\\_Page](http://wiki.openoffice.org/wiki/Main_Page) (Дата обращения 30.06.2013).
9. Справочная информация по модулям Apache OpenOffice. Режим доступа: <http://www.openoffice.org/api/docs/common/ref/com/sun/star/module-ix.html>. (Дата обращения 06.07.2013)
10. Apache OpenOffice Developer's Guide. Режим доступа: [http://wiki.openoffice.org/wiki/Documentation/DevGuide/OpenOffice.org\\_Developers\\_Guide](http://wiki.openoffice.org/wiki/Documentation/DevGuide/OpenOffice.org_Developers_Guide). (Дата обращения 3.07.2013)
11. Питоньяк Э. Полезная информация по макросам для OpenOffice: пер. с англ. - Режим доступа: [http://www.pitonyak.org/AndrewMacro\\_rus.odt](http://www.pitonyak.org/AndrewMacro_rus.odt) [Useful Macro Information For OpenOffice By Andrew Pitonyak] (Дата обращения 22.07.2013)
12. Информационная управляющая система МГТУ им. Н. Э. Баумана "Электронный университет". Концепция и реализация / Агеева Т. И., Балдин А. В., Барышников В.А. [и др.] ; ред. Федоров И. Б., Черненький В. М. М. : Изд-во МГТУ им. Н. Э. Баумана, 2009. 374 с.
13. Справка по формату XML. Режим доступа: <http://technet.microsoft.com/ru-ru/library/ms177276.aspx> (Дата обращения 22.10.2013)

14. Документация по формату RTF. Режим доступа: [http://www.delphi-club.ru/delphi/rtf\\_format.html](http://www.delphi-club.ru/delphi/rtf_format.html). (Дата обращения 22.10.2013)
15. Справка по формату CSV. Режим доступа: <http://ru.wikipedia.org/wiki/CSV>. (Дата обращения 22.10.2013)
16. Справка по формату XSL. Режим доступа: <http://ru.wikipedia.org/wiki/XS> (Дата обращения 22.10.2013)
17. Документация по формату ODF. Режим доступа: <http://www.gost-load.ru/Index/50/50328.htm>. (Дата обращения 22.10.2013)
18. Справка по формату HTML Режим доступа: <http://ru.wikipedia.org/wiki/Html> (Дата обращения 22.10.2013)
19. Начала программирования в OpenOffice.org: пер. с англ. - М.: Инфра-Ресурс, 2007. – 37 с. [OpenOffice.org Developer Guide – Ch.2 First Steps]
20. Справочная информация по созданию шлюзов через UNO. Режим доступа: [http://wiki.openoffice.org/wiki/Uno/Article/About\\_Bridges](http://wiki.openoffice.org/wiki/Uno/Article/About_Bridges) (Дата обращения 05.07.2012)
21. Справочная информация по программированию на языке C++ по технологии UNO. Режим доступа: <http://www.openoffice.org/api/docs/cpp/ref/names/index.html>. (Дата обращения 05.07.2012)
22. Справочная информация по программированию на языке Java по технологии UNO. Режим доступа: <http://www.openoffice.org/api/docs/java/ref/overview-summary.html>. (Дата обращения 05.07.2012)