

УДК 004.624

ТЕХНОЛОГИЯ И ПРОГРАММНАЯ БИБЛИОТЕКА ДЛЯ ЭКСПОРТА ТАБЛИЦ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ В ФОРМАТЕ XML

*Живых С.Ю., студент
кафедры «Система обработки информации и управление»
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана*

*Научный руководитель: Виноградов В.И., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
chernen@bmstu.ru*

В представленной работе рассматриваются принципы экспорта и импорта данных в формате XML. Для промежуточного хранения данных все чаще используется формат XML, так как эта технология обладает массой преимуществ, важнейшими из которых являются гибкость представления данных и простота обработки информации. Данные представляются в виде иерархической структуры, что обеспечивает наглядность их представления. Информацию, которая хранится в этой структуре, легко преобразовать в HTML и XHTML документы для последующего опубликования на web-страницах.

Экспорт данных в структуру XML может быть осуществлен из любого структурированного источника данных. Чаще всего источником экспорта является база данных.

Обработка структур XML бывает следующих видов:

- Обработка документов XML при помощи событий.
- Обработка документов при помощи DOM.

Событийную обработку выполняют программы, называемые процессорами XML. Они предоставляют набор функций для текстовой обработки XML документа и последующего импорта данных из него в другие форматы. Основным преимуществом данного подхода является то, что данный метод требует мало ресурсов. Недостатком - то, что документы просматривается только один раз, от начала к концу.

Объектная модель документа (Document Object Model, DOM) дает пользователю возможность манипулировать содержанием документа XML, так как делает его доступным для программы и сценария. Обработчик документов, основываясь на DOM, формирует дерево элементов, которые являются тэгами в документе XML. Преимуществом данного подхода является то, что в процессе обработки можно изменять

содержимое и добавлять новые элементы. Недостатком — то, что данный метод требует больше времени для обработки и больше ресурсов [1].

В каждом из рассмотренных подходов можно воспользоваться стандартными технологиями конвертации или написать свой собственный синтаксический анализатор. «Синтаксический анализ - это процесс сопоставления линейной последовательности лексем языка с его формальной грамматикой.» «Синтаксическим анализатором называют программу или часть программы, выполняющей синтаксический анализ» [2].

На предприятии часто возникает потребность в формировании отчета в формате HTML. Для этого разработано большое количество программных средств [3]. Недостатками этих систем являются сложность пользовательского интерфейса и отсутствие гибкости в формировании представления данных. Также в этих системах не всегда встречается поддержка импорта данных из базы данных. Поэтому предлагается разработать систему классов для формирования отчетов в формате HTML. Данные отчета должны извлекаться из базы данных на основе SQL-запроса произвольной структуры и сохраняться в файле формата XML. Документ XML содержать сами данные и информацию о полях таблиц, откуда они извлечены..

Необходимо создать программу, предназначенную для формирования отчетов в формате HTML из данных, полученных из базы данных путем формирования пользователем одного или нескольких SQL запросов к ним. Результат выборки сохраняется в промежуточном XML файле, который содержит результаты запроса, а также дату и время создания, и прочие необходимые параметры. После выбора пользователем файла с XSLT шаблоном должен быть создан HTML отчет для последующего просмотра и печати.

Структура XML была разработана для промежуточного хранения результатов выполнения SQL-запросов. В таблице приведены элементы структуры XML в нотации XPath.

Название поля	Объект в структуре XML в нотации XPath
Дата\время создания	/root/dt_create
Комментарий	/root/comment
Результат запроса	/root/ branch
Содержимое всех строк	/root/ branch /rows
Содержимое всех колонок	/root/ branch /columns
Название i-го столбца	/root/ branch /columns/ column

Содержимое i-ой колонки	/root/ branch /rows/ row
Название i-го столбца на русском языке	/root/ branch /columns/ column/ @rus
Длина данных в ячейке i-го столбца	/root/ branch /columns/ column / @len

XML-документ содержит следующие сведения:

Дата\время создания (dt_create) – дата и время создания документа XML пользователем.

Комментарий (comment) – примечания пользователя к выборки из базы данных.

Результат запроса (branch) – результат SQL запроса к базе данных.

Содержимое всех колонок (columns) – содержит имена из столбцов таблиц(ы), к которым(ой) был сформирован SQL запрос.

Содержимое всех строк (rows) – содержимое всех ячеек из строки таблиц(ы), к которым(ой) был сформирован SQL запрос.

Название i-го столбца (column) – название одного из столбцов таблиц(ы), к которым(ой) был сформирован SQL запрос.

Содержимое i-ой колонки (row) – содержимое одной ячейки из строки таблиц(ы), к которым(ой) был сформирован SQL запрос.

Название i-го столбца на русском языке(@rus) – перевод названия столбца на русский язык, заполняется пользователем после обработки.

Максимальная длина данных в ячейке i-го столбца(@len) –длина поля в столбце в базе данных, из которого берутся данные.

Структура XML документа показана на рисунке 1:

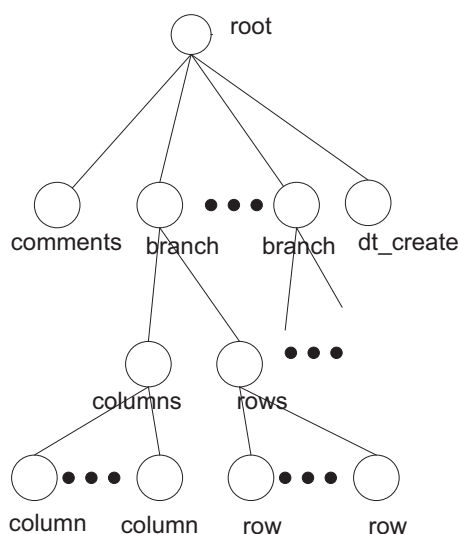


Рис. 1. Структурная схема DOM документа

Существует ряд XSLT шаблонов для обработки этой структуры, которые заранее сформированы и хранятся на жестком диске пользователя или сетевом ресурсе. Шаблон содержит скрипт, с помощью которого можно обработать заранее известную структуру XML файла и преобразовать данные из этого файла в данные в формате HTML.

Для выполнения экспорта и импорта XML был разработан класс ГенераторОтчетов, который содержит в себе набор методов, решающих эту задачу. Экспорт данных в структуры XML происходит по схеме, представленной на рисунке 2.

Пользователь составляет SQL-запросы и вводит комментарии к будущему отчету. Программа выполняет эти запросы и на основе полученных результатов формирует XML-документ. При этом дата и время выбираются из системных параметров.

Импорт данных из структуры XML осуществляется в структуру HTML.

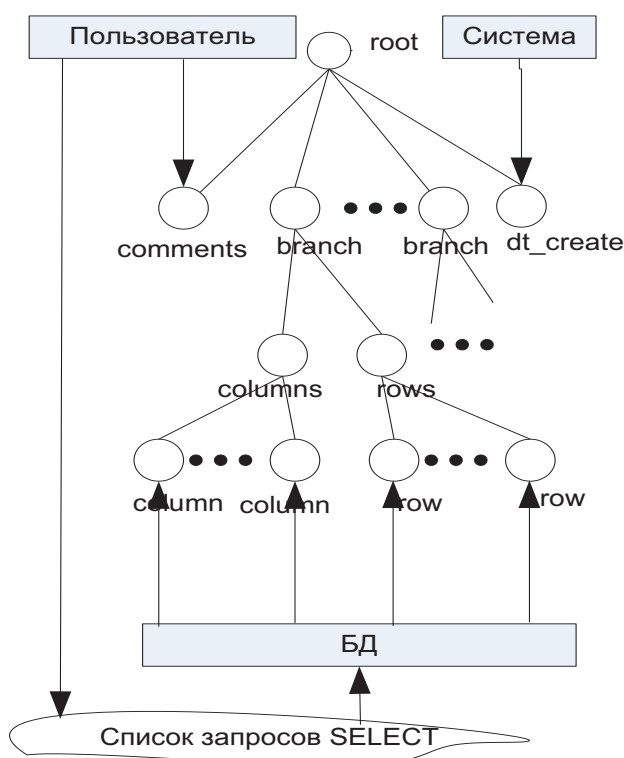


Рис. 2. Схема экспорта данных в структуру XML

Для реализации генератора отчетов предложен следующий список функций:

- Функция создания и сохранения файла XML документа.
- Функция создания DOM документа из списка запросов.
- Функция выполнения запроса и помещения результата в ветку дерева XML документа.
- Функция генерации Html документа на основе XML файла и XSLT шаблона.
- Функция генерации Html документа на основе объекта класса Отчет.

Разделим эти функции на два раздела: функции экспорта XML и функции импорта XML. Далее будут рассмотрены функции экспорта XML.

Функция создания и сохранения файла XML документа. Краткое название функции на русском «создание_файла_XML», на английском - «create_file_XML». На вход функции поступает список объектов класса Запрос и имя файла XML. Функция возвращает результат выполнения (булевский).

При вызове этой функции перед пользователем открывается диалог сохранения файла на жесткий диск, в котором он может выбрать имя файла с расширением xml. После выбора имени файла функция создает DOM дерево и вызывает функцию Создание_документа_XML, в которую передает указатель на DOM документ. В результате успешного выполнения пользователю возвращается true, иначе false.

Функция создания DOM документа из списка запросов. Краткое название функции на русском «создание_документа_XML», на английском «create_DOM_document». На вход функции поступают список объектов класса Запрос и указатель на DOM документ. Функция возвращает результат выполнения (булевский).

Данная функция выполняет формирование DOM документа, содержащего два узла: Comments – комментарии пользователя к формируемому DOM дереву данных, и dt_create – текущее время системы. Одним из параметров этой функции является список запросов к базе данных пользователя, каждый из которых последовательно обрабатывается в цикле при вызове функции Создание_ветки_XML. В результате успешного выполнения пользователю возвращается true, иначе false.

Функция выполнения запроса и помещения результата в ветку дерева XML документа. Краткое название функции на русском «создание_ветки_XML», на английском «create_branch_XML». На вход функции поступают объект класса Запрос и указатель на DOM документ. Функция возвращает результат выполнения (булевский).

Данная функция выполняет запрос, который является параметром функции, к базе данных пользователя. Из результата выполнения запроса формируется ветвь DOM документа, которая сразу прикрепляется к нему. Ветвь показана на рисунке 3. В результате успешного выполнения пользователю возвращается true, иначе false.

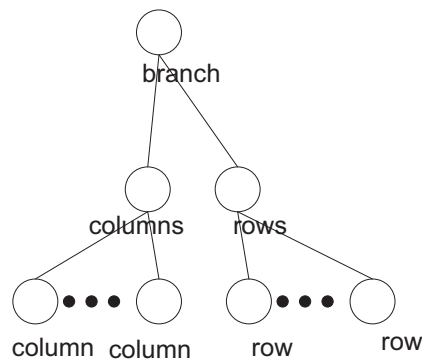


Рис. 3. Ветвь DOM дерева

Далее рассмотрены функции импорта XML. Функция генерации Html документа на основе XML файла и XSLT шаблона. Краткое название функции на русском «Xml_и_Xslt_в_Html», на английском «Xml_and_Xslt_to_XML». На вход функции поступают имя файла XML, имя файла XSLT шаблона и имя файла HTML отчета. Функция возвращает результат выполнения (булевский).

В функции имеется три параметра. Первый из которых имя файла XML, хранящегося на жестком диске пользователя или сетевом ресурсе, который содержит данные из пользовательской базы данных. Второй - имя XSLT шаблона, содержащего текст шаблона обработки XML файла. XSLT шаблон также хранится на жестком диске пользователя или на сетевом ресурсе. Третий параметр - имя HTML документа, который указывается в диалоге сохранения HTML документа на жестком диске. Функция формирует HTML документ и заполняет его результатами слияния XML и XLSX файлов, которое выполняется стандартными библиотеками. В результате успешного выполнения пользователю возвращается true, иначе false.

Функция генерации Html документа на основе объекта класса Отчет. Краткое название функции на русском «создать_отчет», на английском «create_order». На вход функции поступают объект класса Отчет и имя файла HTML отчета. Функция возвращает результат выполнения (булевский).

Из объекта класса отчета извлекается требуемая информация для создания отчета, с помощью остальных функций этого класса документ отчета формируется и сохраняется в файл HTML формата.

Такая схема разбиения на функции удобна, поскольку при изменениях структуры XML, нужно будет не переписывать все функции, а изменить лишь одну, а именно «Создание_ветки_XML». Все остальные функции не зависят от структуры XML, что обеспечивает гибкость разработанному классу функций.

Для программной реализации генератора отчета была разработана система классов, показанная на рисунке 4.

Класс АбстрСущность был разработан как абстрактный класс для нескольких объектов, которые соответствовали отдельным таблицам в БД. Один объект соответствует одной строке в одноименной таблице базы данных. Класс содержит один атрибут Id и методы для взаимодействия с ним.

Класс Запрос представляет собой наследника класса АбстрСущность. Содержит три атрибута: имя_запроса, текст запроса и описание_запроса. Нужен для хранения данных и выполнения запросов. Методы этого класса служат для установки и изменения значений его атрибутов при работе с объектами этого класса.

Класс Шаблон представляет собой наследника класса АбстрСущность. Содержит три атрибута: файл, описание_шаблона и формат_шаблона. Поле файл содержит локальный абсолютный путь к файлу в системе пользователя. Нужен для хранения документа шаблона. Методы этого класса служат для установки и изменения значений его атрибутов при работе с объектами этого класса.

Класс Отчет представляет собой наследника класса АбстрСущность. Содержит шесть атрибутов: название_отчета, описание_отчетов, дата_время_создания, шаблон_отчета, формат_отчета и список_запросов. Поле файл содержит локальный абсолютный путь к файлу в системе пользователя. Нужен для хранения данных и последовательного создания отчета. Методы этого класса служат для установки и изменения значений его атрибутов при работе с объектами этого класса.

Вся информация с помощью которой пользователь создает HTML отчеты, должна сохраняться в собственной базе данных программы. Это информация о запросах, отчетах, шаблонах обработки данных и форматах этих данных. В соответствии с этим были разработаны четыре класса объектов с аналогичными именами; они были унаследованы от абстрактного класса АбстрСущность, который содержал один лишь атрибут id записи. Класс ГенераторОтчетов взаимодействует с классом Запрос и Отчет. Класс Отчет содержит ссылки на объекты классов Шаблон и Запрос. Эти объекты обрабатываются с помощью функций класса ГенераторОтчетов, а результат преобразуется в формат XML для дальнейшей работы с данными.

Система классов была реализована на языке C++ в среде Qt при реализации модуля формирования отчетов при разработке АСУ предприятия. Система классов при необходимости может работать не только с форматами XML, XSLT и HTML. Дописав функции обработки других форматов в класс ГенераторОтчетов, можно получить

совокупность методов, которая позволит формировать отчеты по данным из базы данных в любом популярном формате.

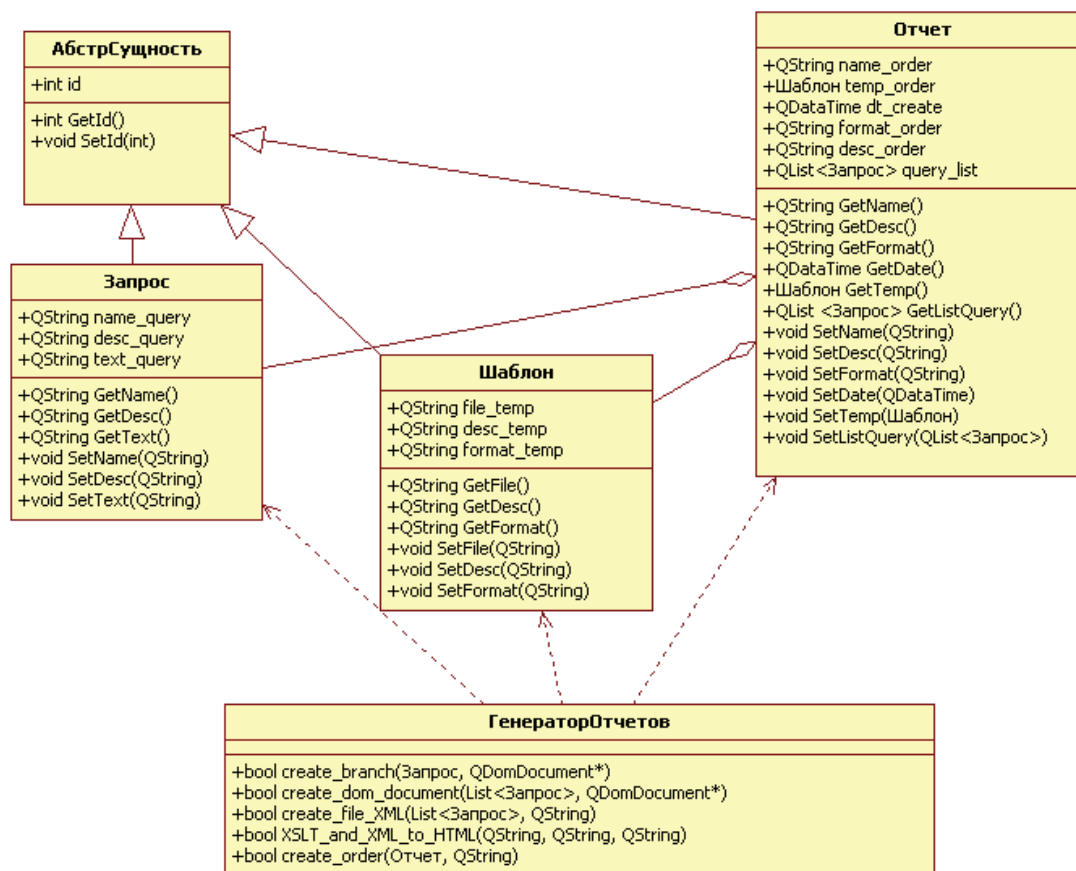


Рис. 4. Диаграмма классов библиотеки экспорта данных в формате XML

Ниже приведен пример работы с описанной библиотекой классов:

Шаг 1.Получение текущей даты и времени

```
QDateTime dt=QDateTime.currentDateTime();
```

Шаг 2.Создание объекта Шаблон с локальной ссылкой на файл

```
Template* temp= new Template("C:/temp.xml","XSLT","Шаблон для отчета №2");
```

Шаг 3.Формирование списка объектов Запрос

```
QList <Query*> qlist;
```

```
Query* query1=new Query("Запрос1","Запрос к таблице 1","SELECT * FROM table1");
```

```
Query* query2=new Query("Запрос2","Запрос к таблице 2","SELECT * FROM table2 WHERE col>100");
```

```
qlist.append(query1);
```

```
qlist.append(query2);
```


Шаг 4. Создание объекта Отчет с заданным Шаблоном, списком Запросов и прочими характеристиками.

```
Paper* paper= new Paper("HTML",temp,dt,"Отчет №2","Отчет содержит информацию о таблице 1 и таблице 2",qlist);
```

Шаг 5. Открытие диалога создания файла html, для сохранения результата выполнения функции.

```
QString file;  
file = QFileDialog::getSaveFileName(this, tr("Создание отчета"), "",  
tr("Html отчет (*.html);;All Files (*.*)"));  
PaperGenerator::paperToHtml(paper, file);
```

Данный текст сформирует HTML отчет по выборкам из таблицы 1 и таблицы 2 на основании шаблона XSLT с локальным путем [C:/temp.xsl](#). Результат сохраняется в указанный пользователем файл [4].

В работе были освещены основные этапы разработки алгоритмов и структур данных для создания генератора отчетов в формате HTML с промежуточным хранением результатов в формате XML. Данный генератор разработан для того, чтобы пользователь, подключившись к собственной базе данных и сформировав к ней несколько запросов типа Select, легко мог создать отчет в формате HTML, имея подготовленный XSLT -шаблон для преобразования. Возможно расширение библиотеки путем добавления новых функций преобразования. Таким образом, разработанный генератор отчетов может быть использован как в коробочном варианте, так и в качестве каркаса для создания собственного генератора для необходимых форматов данных. В работе приведены примеры использования разработанной технологии и библиотеки классов для решения производственной задачи коммерческого предприятия.

Список литературы

1. Леоненков А. Самоучитель UML. Серия "Самоучитель". - СПб.: BHV-Санкт-Петербург, 2002. 304 с.
2. Спенсер П. XML. Проектирование и реализация. М.: Лори, 2001. 510 с.
3. Белоусова В.И., Игушева Ю.С. Классификация источников данных и создание унифицированной модели представления данных / Проблемы построения и эксплуатации систем обработки информации и управления: Сб. статей под ред. Черненко В.М. - М.:Издательство МГТУ им. Н.Э.Баумана. - 2011 г., Выпуск 9. - с.44-56.

4. Бланшет Ж., Саммерфилд М. C++ GUI Programming with Qt 4. М.: КУДИЦ-Пресс, 2008. 718 с.
5. Рамбо Дж., Блах М. Объектно-ориентированное моделирование и разработка. М.: Питер, 2007. 544 с.