

УДК 004.9

## РАЗРАБОТКА АЛГОРИТМА ЧТЕНИЯ ФАЙЛОВ БАЗЫ ДАННЫХ В СИСТЕМЕ ИНТЕРАКТИВНОГО УЧЕТА ВНЕБЮДЖЕТНЫХ СРЕДСТВ ВУЗА

©*Саетова Л. И., Казанский национальный исследовательский технологический университет, г. Нижнекамск, Россия, saetova\_stud@mail.ru*

©*Галеев Э. Р., канд. техн. наук, Казанский национальный исследовательский технологический университет, г. Нижнекамск, Россия, eldargaleev@inbox.ru*

## DEVELOPMENT OF ALGORITHM OF READING DATABASE FILES IN THE SYSTEM OF INTERACTIVE ACCOUNTING OF EXTRA BUDGETARY FUNDS GIVEN OF THE UNIVERSITY

©*Saetova L., Kazan National Research University of Technology,  
Nizhnekamsk, Russia, saetova\_stud@mail.ru*

©*Galeev E., Ph.D., Kazan National Research University of Technology,  
Nizhnekamsk, Russia, eldargaleev@inbox.ru*

*Аннотация.* Автоматизация деятельности высших учебных заведений является актуальной задачей в настоящее время. Осуществление информационной поддержки необходимо для эффективного управления не только учебным процессом, но и всей финансово–хозяйственной деятельностью вуза. Одной из подзадач в бухгалтерском учете высшего учебного заведения является учет внебюджетных средств от образовательной деятельности. Автоматизацию этого процесса можно осуществлять различными способами, применяя разнообразные программные продукты. Современным решением может стать разработка веб–приложения, работающего в локальной сети вуза.

Данная статья рассматривает разработку алгоритма чтения файлов базы данных формата DBF для использования полученной информации с целью построения отчетов, публикуемых в веб–приложении системы интерактивного учета оплаты обучения в высшем учебном заведении на примере Нижнекамского химико–технологического института. Ставится задача интеграции используемой в институте базы данных, состоящей из набора таблиц формата DBF, в разрабатываемое веб–приложение.

Описывается реализованный в институте способ автоматизации учета внебюджетных средств от образовательной деятельности и рассматриваются недостатки такого подхода. Приводится структура файла формата DBF, на основе которой разрабатывается процедура доступа к файлам базы данных из веб–приложения.

Приводится условная схема, отражающая связи между классами и объектами программы во время выполнения очередного запроса к данным базы данных.

*Abstract.* Nowadays automation of the work of higher education institutions is an important task. Information support is necessary for effective management of not only educational process, but whole financial and economic activities of the university. One of the subtasks of accounting of higher educational institutions is an accounting for extra–budgetary funds given from educational activities. Automation of this process can be implemented in different ways using various software.

One of the actual solutions is the development of web–application working in a local network of the university. This article reveals the development of algorithm of reading database's files in DBF format for using received information with the aim of making reports in web–application of

the system of interactive accounting of extra-budgetary funds given of the higher educational institution in case of Nizhnekamsk Institute of Chemical Technology. The problem is to integrate database used in the institute, which consists of a set of DBF tables, into the developed web-application.

The article describes the method for automating the accounting of extra-budgetary funds from educational activities, implemented at the institute, and the drawbacks of this approach. It also reviews the structure of the DBF file, on the basis of which the procedure for accessing database files from a web-application is developing.

An article shows a conditional diagram that represents connections between the classes and objects of the program during the execution of the query to the database data.

*Ключевые слова:* база данных, веб-приложение, dbf-файл, алгоритм чтения таблиц, бухгалтерский учет, автоматизация, вуз.

*Keywords:* data base, web-application, dbf-file, algorithm of tables reading, accounting, automation, university.

В современных условиях развития информационных технологий, предлагающих широкое многообразие программного и технического обеспечения и различных инструментов проектирования и программирования, актуальным является вопрос о целесообразности автоматизации тех или иных видов деятельности организации, если она обладает обширной информационной системой с множеством информационных потоков между структурными единицами.

Высшее учебное заведение, как и любое предприятие, представляет собой организацию с иерархической структурой, в состав которой входят различные отделы, выполняющие те или иные функции. Автоматизация любого из подразделений значительно облегчает и ускоряет работу сотрудников.

#### *Постановка задачи исследования*

Рассмотрим бухгалтерский учет вуза на примере Нижнекамского химико-технологического института (НХТИ). Для этой цели в институте используется система автоматизации «Парус-Бюджет 7», представляющая собой программный продукт, построенный на базе архитектуры системы файл-сервер (Рисунок 1) с использованием СУБД FoxPro [1].

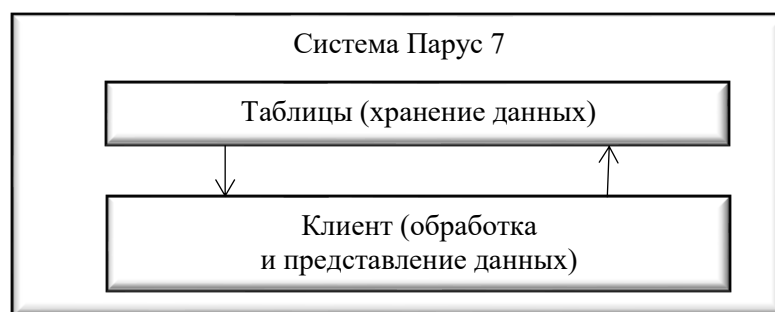


Рисунок 1. Архитектура файл-сервер.

Под учет оплаты за обучение студентами адаптирован один из модулей системы «Склад». В нем ведется список обучающихся студентов, их лицевых счетов, выставляемых им исходящих счетов на оплату и журнал учета оплаты за обучение на основе реестров, предоставляемых банком. Такой способ автоматизации системы учета внебюджетных средств ВУЗа имеет как свои достоинства, так и недостатки. Плюсом является то, что большая часть функций отдела автоматизирована, есть наработанная более чем за 10 лет база данных со списком всех когда-либо обучавшихся за этот период студентов, их платежей и выставляемых им счетов на оплату. Недостаток такого способа автоматизации заключается в том, что модуль не специализирован под конкретные задачи отдела, а лишь адаптирован под выполнение некоторых из его функций. Как следствие, в модуле нет встроенных специализированных отчетов для учета оплаты. Например, для формирования сводной таблицы о задолженности по всему институту необходимо отдельно собрать данные о задолженности по каждому факультету и форме обучения.

В связи с этим в институте разрабатывается веб-приложение системы интерактивного учета внебюджетных средств, которое позволит получать актуальную информацию об оплате, необходимую всем деканатам института, главному бухгалтеру, директору и бухгалтеру, ведущему учет.

Важным элементом проектируемой автоматизированной системы учета является разработка процедуры чтения файлов БД, которая, в свою очередь, ежедневно наполняется новыми данными через систему «Парус 7 Бюджет».

#### *Алгоритм чтения таблиц базы данных для формирования отчетов*

База данных системы «Парус 7 Бюджет», используемая для ведения учета оплаты, представляет собой множество dbf-файлов, в которых хранится вся информация (1–4).

Все файлы формата DBF относятся к форматам хранения данных, которые используются в роли одного из нескольких стандартных методов хранения и передачи информации посредством систем управления базами данных, электронных таблиц и так далее. Файл формата DBF состоит из:

- заголовка фиксированной длины, в котором хранится информация о структуре базы данных и количестве допустимых записей;
- списка дескрипторов полей (описание структуры записей). Число дескрипторов — переменная величина. Для отделения поля дескрипторов от записей файла используется терминальный байт (символ) — символ перевода строки;
- непосредственно тело файла, состоящее из последовательно организованной таблицы с записями фиксированной длины. Каждая запись содержит заголовочный байт, помеченный символом пробела. Записи, помеченные как удаленные, отмечаются символом звездочки (\*).

Для чтения таблицы базы данных формата DBF необходимо побайтово считывать заголовок файла и дескрипторы полей, которые содержат информацию о полях и записях в файлах. Полученные метаданные позволят при необходимости считывать все записи файлов. С помощью метаданных в дальнейшем происходит конфигурация класса DbfReader, который производит чтение таблиц БД. Зная размер и структуру метаданных, а также размер и структуру записей, можно прочитать любую запись в dbf-файле.

Пусть  $m$  — размер метаданных,  $s$  — размер записи. Для того чтобы считать запись с индексом  $i$ , сначала необходимо пропустить первые байты, образующие метаданные, размер которых определяется следующим образом:  $m=s \times (i-1)$ . После этого происходит чтение записи размером  $s$  байт.

Класс DbfReader позволяет последовательно считывать записи таблицы и получать типизированные значения их полей. Чтение файла таблицы в приложении производится с помощью классов FileStream и BinaryReader из пространства имен System.IO.

Класс FileStream предоставляет возможность для чтения из файла и записи данных в файл. Он позволяет работать как с текстовыми, так и с бинарными файлами, поддерживает синхронные и асинхронные операции ввода/вывода.

Класс BinaryReader предоставляет методы, которые упрощают чтение примитивных типов данных из бинарного потока. Его удобно использовать, когда известна структура файла. Соответствующий типу данных метод считывает данные определенного типа и перемещает указатель на размер этого типа в байтах. Например, чтобы прочитать следующий байт как значение типа Boolean, используется метод BinaryReader.ReadBoolean(), который производит чтение, возвращает значение соответствующего типа и передвигает позицию в потоке на один байт. Для чтения данных типа Char используется метод BinaryReader.Char(), который считывает один символ и перемещает указатель на столько байтов, сколько занимает символ в текущей кодировке (2).

Чтение таблицы производится согласно следующему алгоритму:

Файл открывается в режиме чтения:

```
var stream = new FileStream(path, FileMode.Open, FileAccess.Read).
```

Затем полученный ранее байтовый поток и кодировка файла передаются конструктору класса BinaryReader:

```
BinaryReader = new BinaryReader(stream, encoding, false).
```

Далее производится чтение заголовка dbf-файла:

```
Header = new DbfHeader(BinaryReader).
```

Производится чтение дескрипторов полей:

```
Columns = ReadColumns(BinaryReader).
```

Затем производится переход непосредственно к чтению записей:

```
SkipToFirstRecord(BinaryReader).
```

Вышеприведенные операции необходимы для того, чтобы сконфигурировать класс DbfReader. После того как DbfReader будет сконфигурирован, происходит чтение всей таблицы.

Для этого используется цикл while. Оператор while выполняет оператор или блок операторов, пока определенное выражение не примет значение false. Таки образом, передавая в качестве выражения результат, возвращаемый методом DbfReader.Read(), в теле цикла можно получать значения полей считанной записи. В свою очередь, метод DbfReader.Read() производит попытку чтения записи и возвращает значение true до тех пор, пока не будет достигнут конец таблицы, иначе метод возвратит результат.

```
false.
```

```
while (dbfDataReader.Read())
```

```
{
```

```
    if (dbfDataReader.DbfRecord.IsDeleted)
```

```
    {
```

```
        continue;
```

```
    }
```

```
    ret.Add(GetValues(dbfDataReader));
```

```
}
```

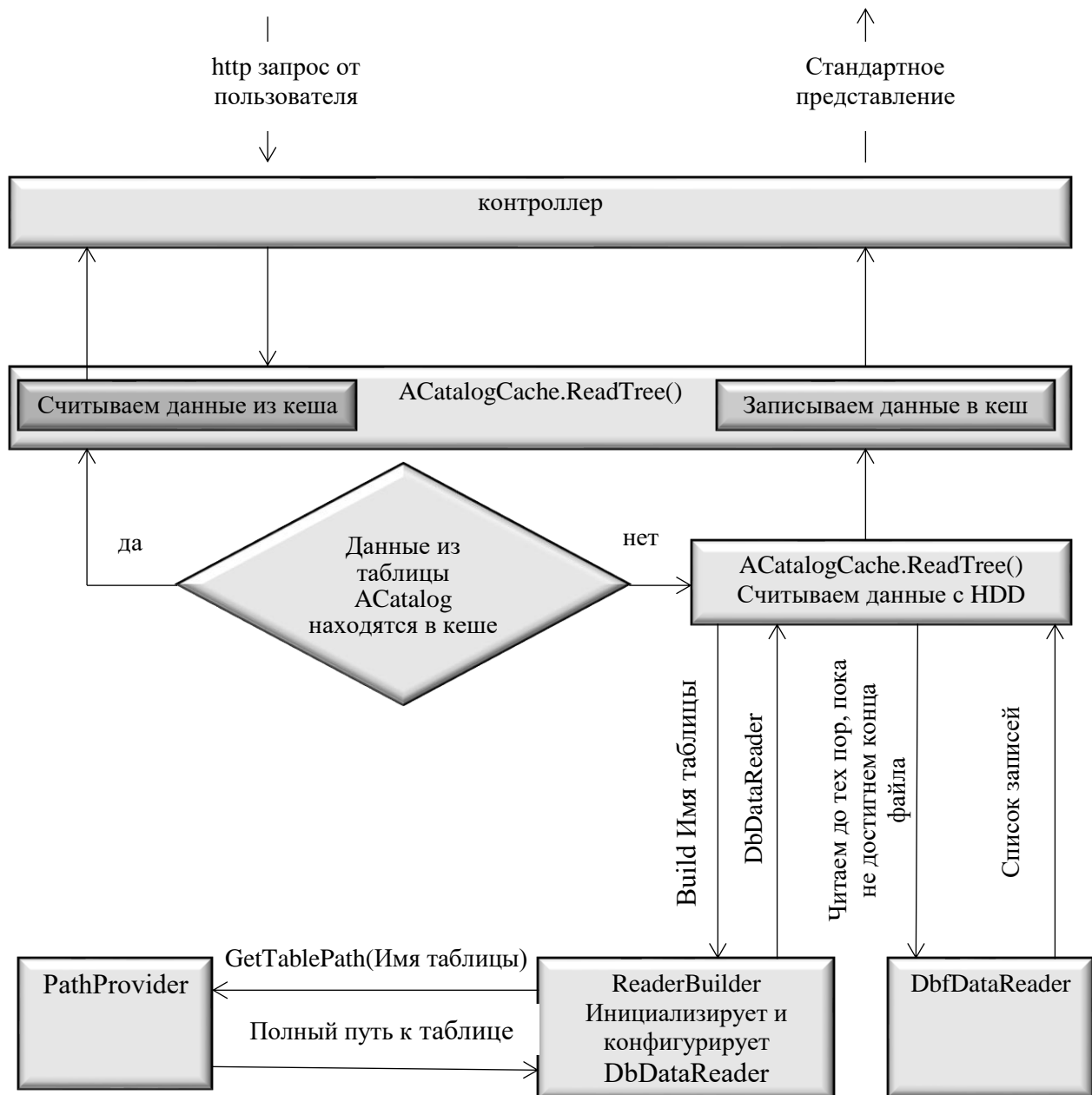


Рисунок 2. Процесс получения данных из таблиц БД.

*Процесс обращения к таблицам БД из веб-приложения*

Рассмотрим связи, протекающие внутри веб-приложения при обращении к какой-либо таблице базы данных на примере таблицы ACatalog (Рисунок 2).

Для того чтобы заново не считывать таблицы базы данных при каждом новом запросе, они хранятся в кэше. Кэш хранится в ОЗУ сервера, время обновления данных в кэше можно настраивать, например, через каждые 12 часов.

От пользователя поступает http-запрос данных из таблицы ACatalog. Метод ACatalogCache.ReadTree() проверяет наличие данных таблицы в кэше. Если данные уже находятся в кэше, они передаются в виде визуализированного представления пользователю в виде http-ответа. Если запрашиваемые данные не содержатся в кэше, происходит считывание данных с жесткого диска. Для начала необходимо получить полный путь к таблице. Путь к самой базе данных хранится в файле конфигурации приложения. Класс PathProvider

считывает из конфигурации полный путь к БД и добавляет к нему имя запрашиваемой таблицы. ReaderBuilder создает экземпляр класса DbfDataReader, передавая ему в качестве параметров конструктора полный путь к таблице и выбирая кодировку, соответствующую кодировке таблицы базы данных. В свою очередь, класс ACatalogReader обращается к сконфигурированному экземпляру класса DbfDataReader, что позволяет ему последовательно считывать все записи таблицы вплоть до конца файла. Затем считанные данные записываются в кэш и оттуда передаются в виде визуализированного представления пользователю [1–2].

#### *Заключение*

В результате проделанной работы была разработана процедура чтения таблиц базы данных, которая будет использоваться в разрабатываемой системе интерактивного учета внебюджетных средств НХТИ с целью построения отчетов и их публикации через веб–приложение.

Таким образом, достигнута интеграция веб–приложения с существующей базой данных бухгалтерии института, наполняемой через систему «Парус 7 Бюджет», в которой хранятся все текущие и архивные данные по оплате за обучение студентами, обучающимися по внебюджетному финансированию. Это позволит пользователям приложения получать актуальную информацию по задолженности и следить за ходом оплаты обучения.

#### *Источники:*

- (1). 1Парус. Режим доступа: <http://www.parus.com/products/gov/562/> (дата обращения: 10.08.2017).
- (2). Руководство по ASP.NET Core 2.0. Режим доступа: <https://metanit.com/sharp/aspnet5/> (дата обращения: 16.05.2017).
- (3). Руководство по .NET Core. Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/core/> (дата обращения: 16.05.2017).
- (4). Справочник по HTML. Режим доступа: <http://htmlbook.ru/html> (дата обращения: 24.05.2017).

#### *Sources:*

- (1). 1Parus. Access mode: <http://www.parus.com/products/gov/562/> (reference date: August 10, 2017).
- (2). Guide to ASP.NET Core 2.0. Access mode: <https://metanit.com/sharp/aspnet5/> (application date: May 16, 2017).
- (3). Guide to .NET Core. Access mode: <https://docs.microsoft.com/en-us/dotnet/core/> (date of circulation: 16/05/2017).
- (4). Handbook of HTML. Access mode: <http://htmlbook.ru/html> (date of circulation: 24/05/2017).

#### *Список литературы:*

1. Макфарланд Д. Новая большая книга CSS. СПб.: Питер, 2016. 720 с.
2. Рейсиг Д. JavaScript. Профессиональные приемы программирования. СПб.: Питер, 2008. 352 с.

#### *References:*

1. McFarland, D. (2016). A new large CSS book. St. Petersburg, Peter, 720.

2. Resig, J. (2008). JavaScript. Professional methods of programming. St. Petersburg, Peter, 352.

*Работа поступила  
в редакцию 22.05.2018 г.*

*Принята к публикации  
27.05.2018 г.*

---

*Ссылка для цитирования:*

Саева Л. И., Галеев Э. Р. Разработка алгоритма чтения файлов базы данных в системе интерактивного учета внебюджетных средств вуза // Бюллетень науки и практики. 2018. Т. 4. №6. С. 205-211. Режим доступа: <http://www.bulletennauki.com/saetova> (дата обращения 15.06.2018).

*Cite as (APA):*

Saetova, L., & Galeev, E. (2018). Development of algorithm of reading database files in the system of interactive accounting of extra budgetary funds given of the university. *Bulletin of Science and Practice*, 4(6), 205-211.