



ISSN 1998-7927

ВІСНИК

**Східноукраїнського
національного
університету
імені ВОЛОДИМИРА ДАЛЯ**

НАУКОВИЙ ЖУРНАЛ

**№ 8(179) Ч.2
2012**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ

ВІСНИК

**Східноукраїнського
національного університету
імені ВОЛОДИМИРА ДАЛЯ**

**№ 8 (179) Ч.2
2012**

НАУКОВИЙ ЖУРНАЛ

Луганськ 2012

ВІСНИК

СХІДНОУКРАЇНСЬКОГО
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

№ 8 (179) 2012

НАУКОВИЙ ЖУРНАЛ
ЗАСНОВАНО У 1996 РОЦІ
ВИХІД З ДРУКУ - ВІСІМНАДЦЯТЬ
РАЗІВ НА РІК

Засновник
Східноукраїнський національний
університет імені Володимира Даля

VISNIK

OF THE VOLODYMYR DAHL EAST
UKRAINIAN NATIONAL UNIVERSITY

№ 8 (179) 2012

THE SCIENTIFIC JOURNAL
WAS FOUNDED IN 1996
IT IS ISSUED EIGHTEEN TIMES
A YEAR

Founder
the Volodymyr Dahl East Ukrainian
National University

Журнал зареєстровано
в Міністерстві юстиції України

Свідоцтво про державну реєстрацію
серія КВ № 15607-4079ПР
від 18.08.2009 р.

Registered by the Ministry
of Justice of Ukraine

Registration Certificate
KB № 15607-4079ПР
dated 18.08.2009

Журнал включено до Переліків наукових видань ВАК України (Бюл. ВАК №3 2010 р.), (Бюл. ВАК №5 2010 р.), (Бюл. ВАК №3 2010 р.), (Бюл. ВАК №11 2010 р.), (Бюл. ВАК №7 2011 р.) в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора і кандидата наук з *технічних, економічних, історичних, хімічних та фізико-математичних наук* відповідно.

ISSN 1998-7927

Головна редакційна колегія: Голубенко О.Л., член-кор. Національної академії педагогічних наук, докт. техн. наук (головний редактор), Осенін Ю.І., докт. техн. наук (заступник головного редактора), Смирний М.Ф., докт. техн. наук (заступник головного редактора), Арлінський Ю.М., докт. фіз-мат. наук, Бер Р., докт. техн. наук., професор університету ім. Отто фон Гюріке, Магдебург, Німеччина, Будіков Л.Я., докт. техн. наук., Бузько І.Р., докт. екон. наук, Гадушова З., професор, декан факультету мистецтв університету Філософа Костянтина в Нитрі, Словачія, Галстян Г.А., докт. хім. наук, Голубничий П.І., докт. фіз-мат. наук, Гончаров В.М., докт. екон. наук, Довжук І.В., докт. іст. наук, Житна І.П., докт. екон. наук, Іджер М., докт. техн. наук., професор Познанського технічного університету, Польща, Красовські Е., професор університету природничих наук в Любліні, редактор наукового видання Текі і MOTROLU, Козаченко Г.В., докт. екон. наук, Кондратов С.О., докт. хім. наук, Кудюков Ю.П., докт. хім. наук, Куліков Ю.А., докт. техн. наук, Лазор Л.І., докт. юр. наук, Литвиненко В.Ф., докт. істор. наук, Максимов В.В., докт. екон. наук, Михайлюк В.П., докт. іст. наук, Нагорний Б.Г., докт. соціол. наук, Носко П.Л., докт. техн. наук, Петров О.С., докт. техн. наук, Рач В.А., докт. техн. наук, Рей Р.І., докт. техн. наук, Суханцева В.К., докт. філос. наук, Тюпало М.Ф., докт. хім. наук, Ульшин В.О., докт. техн. наук, Чапка М., докт. екон. наук, професор, іноземний член-кор. Національної академії педагогічних наук України, Польща, Шевченко Г.П., член-кор. Національної академії педагогічних наук України, докт. пед. наук., Хорошко В.О., докт. техн. наук.

Відповідальний за випуск: Статівка Ю.І.

Рекомендовано до друку Вченою радою Східноукраїнського національного університету імені Володимира Даля (Протокол № 10 від 25 травня 2012 р.)

Матеріали номера друкуються мовою оригіналу.

© Східноукраїнський національний університет імені Володимира Даля, 2012
© The Volodymyr Dahl East Ukrainian National University, 2012

ЗМІСТ

ТЕХНІЧНІ НАУКИ

Андрющук О.А., Габараєва К.Р. Проблеми підготовки до Євро-2012.....	6
Бірюков Д.С., Заславський В.А., Сідляренко А.І. Розвиток і впровадження інформаційно-аналітичного забезпечення в сфері керування автодорожнім господарством регіонів	10
Болдирєв О.П., Габараєва К.Р. Проблеми розвитку туризму у Луганській області	16
Войтиков В.А., Статівка Ю.И. Процесс наполнения модели содержательного компонента корпуса текстов на естественном языке	20
Григорович А.Г., Григорович В.Г. Реляційне числення доменів для ненормалізованих відношень	24
Дегтярев А.В., Коваленко В.А., Потапов А.В., Штефан Ю.В. О некоторых особенностях снижения массы трехслойных сотовых каркасов для панелей солнечных батарей.....	30
Дичка І.А., Костенко К.О., Кащинець А.Я. Модифікація методу WPDM для виявлення заданих періодичностей	36
Заміховський Л. М., Клапоуцак О. І. Розробка прогностичних схем підняття рівня паводкових вод	40
Иванов Д.Е. Оценка рассеивания тепла сбис для различных режимов работы	44
Киричевский Р. В., Донченко В. Ю. Математическое моделирование уравнения теплопроводности с двумя пространственными переменными	51
Козленко М. І. Вплив часової форми випадкових сигналів з керованою ентропією на якість обміну даними в автоматизованих системах керування та розподілених комп'ютерних системах	56
Козуб Ю.Г. Прогнозирование долговечности эластомерных виброизоляторов.....	62
Козуб Ю.Г., Козуб Г.О., Жукова В. М. Візуалізація результатів розрахунків в обчислювальному комплексі «Мірела +»	67
Крамаренко Т. А., Козуб Г. А. Этапы создания интерфейса клиентского приложения баз данных MYSQL в виде графического приложения с использованием технологии ADO.net.....	75
Лахно В.А. Управление информационной безопасностью на транспорте в условиях деструктивного воздействия на информационные системы	84
Левицький І.Т. Аналіз вибірки металовключень у сипучій сировині в умовах ВАТ "Керамікбудсервіс"	91
Лобов И.В., Статівка Ю.И. Особенности применения системы обработки голосовых данных	95
Логинов А.В. Анализ процесса разработки медицинских экспертных систем	99

Козуб Ю.Г., Козуб Г.А., Жукова В.Н. ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ РАСЧЕТОВ В ВЫЧИСЛИТЕЛЬНОМ КОМПЛЕКСЕ «MIRELA +»

В статье предоставлен анализ визуализации в системе автоматизированного проектирования конструкций. Предложен новый подход к реализации алгоритмов визуализации результатов расчетов в рамках комплекса «MIRELA +».

Ключевые слова: САПР, метод конечных элементов, конечноэлементная модель, визуализация.

Kozub Y.G., Kozub G.O., Zhukova V. N. VISUALIZATION OF RESULTS OF CALCULATIONS IN CALCULABLE COMPLEX «MIRELA +»

In the article the analysis of visualization is given in a computer-aided of constructions design. The new going is offered near realization of algorithms of visualization of results of calculations within the framework of complex «MIRELA +».

Keywords: CADD, method of eventual elements, konechnoelementnaya model, visualization.

Козуб Юрій Гордійович, к.т.н., доцент, зав.кафедри інженерно-педагогічних дисциплін Луганського національного університету імені Тараса Шевченка.

Козуб Галина Олександрівна, к.т.н., доцент кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка.

Жукова Вікторія Миколаївна, к.п.н., доцент кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка.

Рецензент: Белодедов Виктор Александрович, д.т.н., проф. ВНУ.

Стаття подана 17.06.2012

УДК 004.65

Крамаренко Т. А., Козуб Г. А.

ЭТАПЫ СОЗДАНИЯ ИНТЕРФЕЙСА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ БАЗ ДАННЫХ MYSQL В ВИДЕ ГРАФИЧЕСКОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ADO.NET

В статье рассматриваются этапы создания клиентских приложений баз данных для клиент-серверной СУБД MySQL на языке программирования C# в среде Microsoft Visual Studio на платформе .NET Framework с использованием механизма доступа приложений по технологии ADO.NET.

Ключевые слова: база данных, архитектура клиент-сервер, клиентское приложение, платформа .NET Framework, технология ADO.NET

В современном мире в основе любой информационной системы (ИС) лежит база данных (БД), а точнее система управления базами данных (СУБД). И выбор той или иной СУБД существенно влияет на функциональные возможности информационной системы и проектные решения.

Многопользовательские СУБД позволяют создавать ИС, функционирующие в архитектуре „клиент-сервер“. Пользователь осуществляет работу с базой при помощи приложений БД. Двухуровневые приложения, использующие удаленные базы данных, содержат клиентскую и серверную части.

Выбор разработчиком СУБД усложняется в случае необходимости обеспечить поддержку различных источников данных, причем каждый из таких источников данных может хранить и обрабатывать данные по-своему. Актуальным является выбор языка программирования, т.к. в различных языках программирования разнообразна поддержка работы с той или иной СУБД. То есть, возникает проблема несоответствия обработки информации большинством СУБД и способам обработки информации различными языками программирования.

Решение выдвинутых проблем предлагается в технологии ADO.NET доступа к данным, разработанной компанией Microsoft, и включенной в их платформу .NET

Framework. ADO.NET, как часть Microsoft .NET Framework, представляет собой набор средств и слоев, позволяющих приложению легко управлять и взаимодействовать со своим файловым или серверным хранилищем данных.

Объектно-ориентированный язык программирования C# изначально был создан компанией Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

На сегодняшний день одной из самых популярных клиент-серверных СУБД является MySQL. MySQL свободная СУБД, портирована на большое количество платформ, имеет API (Application Programming Interface, интерфейс прикладного программирования) для языков программирования Delphi, C, C++, Eiffel, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC [1, с. 107].

СУБД MySQL поддерживает библиотеки для языков платформы .NET, что позволяет использовать данную технологию для доступа к базе данных.

Вопросы создания клиентских приложений БД на C# с механизмом доступа к данным по технологии ADO.NET рассмотрены многими отечественными и зарубежными учеными и специалистами в данной области. Так Д. Сеппа подробно исследует возможности объектной модели ADO.NET, в том числе описывает на примерах этапы разработки приложений на языках Visual C# и Visual Basic для взаимодействия с СУБД Microsoft SQL Server, Oracle с использованием ADO.NET [2].

А. Постолит рассматривает технологию Microsoft .NET и базовое инструментальное средство этой технологии Visual Studio .NET., пошагово описывает на двух языках программирования: Visual Basic и C# создание приложений, начиная с формирования пустой формы и заканчивая установкой конечного продукта у клиента, а также описывает особенности базовых объектов доступа к данным и элементов управления и уделяет внимание взаимодействию различных типов приложений с базами данных на примере СУБД Microsoft SQL Server [3, с. 11 – 237].

Курс лекций И. Баженовой знакомит с теоретическими и практическими аспектами разработки распределенных приложений баз данных. Наибольший акцент сделан на приложениях, использующих БД Oracle и Microsoft SQL Server. Подробно изучается применение различных систем программирования и инструментальные средства, используемых для создания распределенных приложений. Рассматривается реализация доступа на базе ODBC, JDBC и OLE DB. В курс введены вопросы применения технологий ADO.NET [4].

В свою очередь П. Шумаков касается вопросов доступа к внешним источникам данных и их хранения в приложении: как работать с данными из приложений, создаваемых на платформе .NET Framework на примере использования в качестве источников данных базы данных Microsoft SQL Server и Microsoft Access [4, с. 13 – 405].

На примере настольного приложения СУБД Microsoft Access и серверного приложения Microsoft SQL Server О. Евсеева и А. Шамшев также знакомят с технологией разработки баз данных средствами ADO на платформе .NET с использованием языка программирования C# [5].

Анализ литературы показал, что вопросы создания клиентских приложений на C# для баз данных MySQL по технологии доступа к данным ADO.NET, требуют дополнительного рассмотрения.

Архитектура клиент-сервер представляет собой вычислительную или сетевую архитектуру, в которой задания или сетевая нагрузка распределены между поставщиками услуг, – серверами, и заказчиками услуг, – клиентами. В системах, построенных по технологии "клиент-сервер", существует два вида интерфейса: интерфейс, реализуемый при помощи клиентского приложения и Web-интерфейс. Интерфейс, реализуемый при помощи клиентского приложения – это компьютерная программа, устанавливаемая на клиентские компьютеры, предназначенная для работы с файлами данных через сеть [1, с. 107 – 108].

В данной статье рассмотрим первый вид интерфейса.

Цель работы – рассмотрение этапов создания клиентских приложений баз данных для клиент-серверной СУБД MySQL на языке программирования C# в среде Microsoft Visual Studio на платформе .NET Framework с использованием механизма доступа приложений по технологии ADO.NET.

ADO.NET – новый этап в технологии ActiveX Data Object (ADO, объекты данных ActiveX), это часть Microsoft .NET Framework, предоставляющая разработчикам набор объектов, позволяющих приложению легко управлять и взаимодействовать со своим файловым или серверным хранилищем данных (рис. 1).

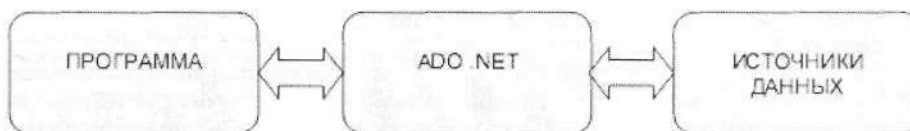


Рис. 1. Место ADO.NET в организации доступа к данным

Среда Microsoft Visual Studio .NET предоставляет для доступа к данным объекты ADO.NET. ADO.NET отличается от предыдущих технологий доступа к данным тем, что она позволяет взаимодействовать с базой данных автономно, с помощью отделенного от базы кэша данных. Автономный доступ к данным необходим, когда невозможно удерживать открытое физическое подключение к базе данных каждого отдельного пользователя или объекта.

Объекты ADO.NET. Все классы .NET группируются в пространства имен. Все функции, относящиеся к ADO .NET, находятся в System.Data. Кроме того, как и любые другие компоненты .NET, ADO.NET работает не изолированно и может взаимодействовать с различными другими компонентами .NET.

Архитектуру ADO.NET можно разделить на две фундаментальные части: подключаемую и автономную. Все классы в ADO.NET можно поделить по этому критерию. Единственное исключение – класс DataAdapter, который является посредником между подключенной и автономной частями ADO.NET.

Поставщики данных .NET. Подключаемая часть ADO.NET представляет собой набор объектов подключений.

Объекты подключений разделяются в ADO.NET по конкретным реализациям для различных СУБД. То есть для подключения к базе данных SQL SERVER имеется специальный класс SqlConnection. Эти отдельные реализации для конкретных СУБД называются *поставщиками данных .NET*. При наличии широкого выбора доступных источников данных ADO.NET должна иметь возможность поддерживать множество источников данных. Каждый такой источник данных может иметь свои особенности или набор возможностей.

Поэтому ADO.NET поддерживает модель поставщиков. *Поставщики для конкретного источника данных* можно определить как совокупность классов в одном пространстве имен созданных специально для данного источника данных. Эта особенность несколько „размыта” для источников данных OleDb, ODBC, так как они по своей сути созданы для работы с любой базой данных совместимых с OLE и ODBC [6, с. 163 – 166].

Так, например, для источника данных БД MySQL существует свой поставщик данных (.NET Framework Data Provider for MySQL), который содержит класс MySQLSqlConnection для подключения к базе данных. Для использования данного провайдера следует дополнительно установить соответствующий драйвер, например, mysql connector net 6.2.2 (или более новой версии).

Подключаемые классы и объекты. В подключаемой части ADO.NET имеются следующие основные классы, предоставляемые поставщиком (провайдером) данных (в скобках указаны классы и объекты для MySQL):

Connection. Этот класс, позволяющий устанавливать подключение к источнику данных (MySQLConnection).

Transaction. Объект транзакций (MySQLTransaction). В ADO.NET имеется пространство имен System.Transaction.

DataAdapter. Это своеобразный шлюз между автономными и подключенными аспектами ADO.NET. Он устанавливает подключение, и если подключение уже

установлено, содержит достаточно информации, чтобы воспринимать данные автономных объектов и взаимодействовать с базой данных (MySqlDataAdapter).

Command. Это класс, представляющий исполняемую команду в базовом источнике данных (MySqlCommand).

DataReader. Это эквивалент конвейерного курсора с возможностью только чтения данных в прямом направлении (MySqlDataReader).

Parameter. Объект параметр команды.

Для доступа и манипулирования данными существуют два пути: 1) использовать класс *DataReader* провайдера данных, позволяющий однонаправленный просмотр данных, режим „только чтение“; 2) использовать класс *DataSet*, реализующий работу с данными, хранимыми в кэше на клиенте.

Класс *DataSet* является ядром автономного режима доступа к данным в ADO.NET. Т.е. объект *DataReader* использует для хранения данных кэш сервера, а объект *DataSet* – кэш клиента.

И так, рассмотрим *этапы создания клиентских приложений баз данных для клиент-серверной СУБД MySQL на языке программирования C# в среде Microsoft Visual Studio на платформе .NET с использованием механизма доступа приложений по технологии ADO.NET* (на примере базы данных *strahovanievar30*, состоящей из трех связанных таблиц и находящейся в папке *data* на сервере MySQL).

В данной работе этап создания интерфейса с элементами (формы, меню, кнопки, переключатели и т.д.) не описывается.

1. Создать проект типа *Windows Form Application* (*Приложение Windows Form*).

2. Настроить соединение с источником данных.

Технология ADO.NET не ориентирована на применение длительных соединений – соединение открывается только на время взаимодействия клиента с источником данных.

Для того чтобы в среде *Microsoft Visual Studio* установить соединение с источником данных, следует или выполнить метод *Open* для объекта соединения, или установить значение свойства *Connection* объекта *Command*. В любом случае следует определить строку соединения.

Строка соединения содержит элементы с минимальной информацией, необходимой для установления подключений, в виде последовательности пар „ключ – значение“ (имя сервера, идентификатор пользователя, пароль, имя БД, кодировка и т.д.). Это можно сделать, используя Мастер настройки источника данных:

команда меню *Data | Add New Data Source* (*Данные | Добавить новый источник данных*);

выбрать источник данных для приложения *Database* (*База данных*);

выбрать (изменить) источник данных на *MySQL DataBase* и соответствующий поставщик данных (рис. 2);

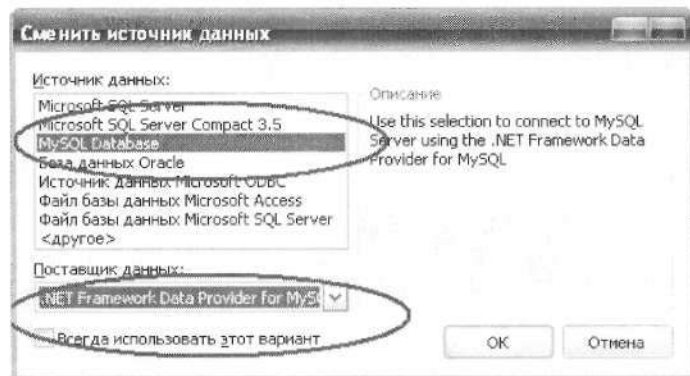


Рис. 2. Диалог выбора источника подключения и поставщика .NET Framework

добавить подключение, установив в поля нужную информацию (т.е. определив содержимое строки соединения) (см. рис. 3 слева);

при необходимости задать дополнительные свойства в строке соединения, например, кодировку (рис. 3 справа);

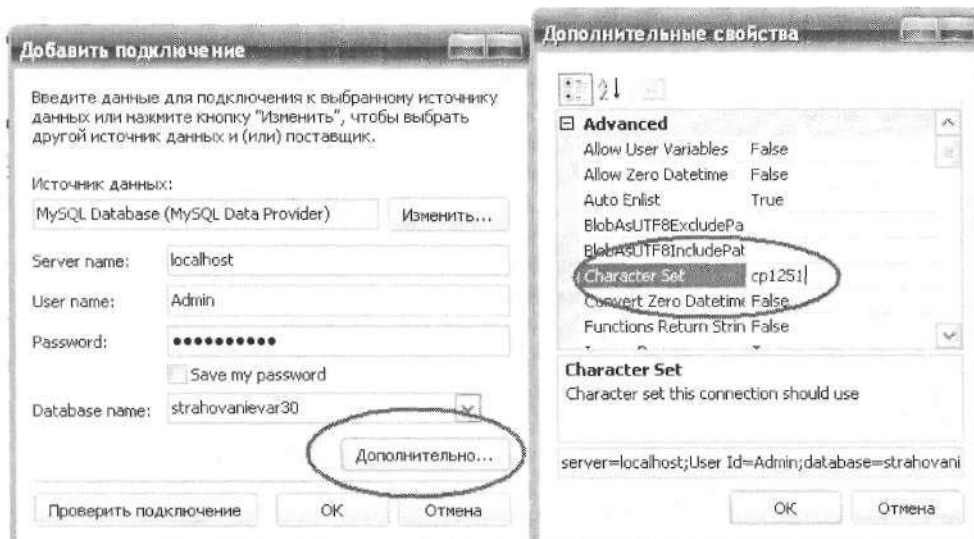


Рис. 3. Диалог настройки строки соединения с сервером БД

проверить корректность созданного подключения к базе данных на сервере, нажав на кнопку *Проверить подключение*;

выбрать включение конфиденциальных данных в строку подключения;

здать имя подключения (здесь *strahovanievar30ConnectionString*);

выбрать объекты базы данных для набора данных (таблицы, хранимые процедуры, представления, функции) и ввести имя набора данных *DataSet* (рис. 4);

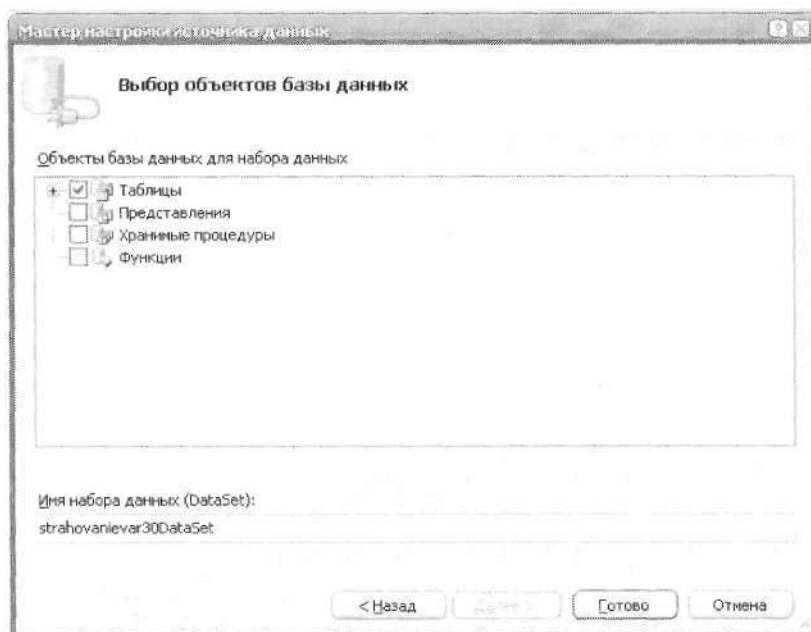


Рис. 4. Диалог выбора объектов БД и набора данных

выбрать в *Solution Explorer* (Обозреватель решений) новый объект *strahovanievar30DataSet.xsd* (схема базы данных) и открыть его в конструкторе (см. рис. 5). В окне *Server Explorer* (Обозреватель серверов) отобразится созданное подключение.

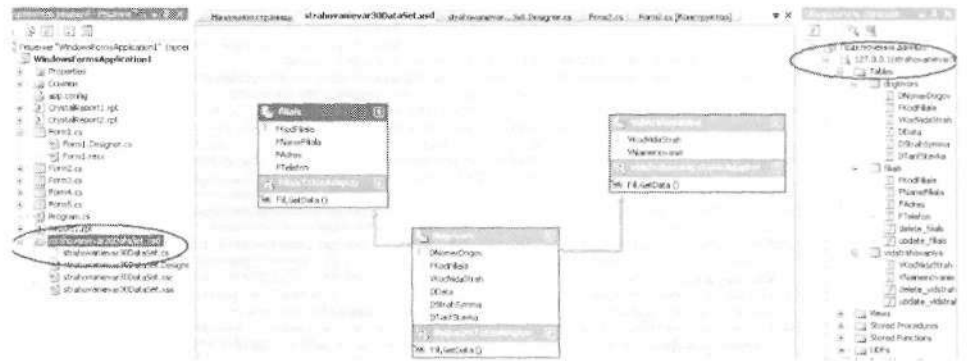


Рис. 5. Пример схемы данных БД strahovanievar30

отредактировать при необходимости в файле проекта *app.config* настройки соединения *ConnectionString*, которое содержит параметры подключения к базе. С указанными ранее настройками соединения выглядит оно так:

```
<connectionStrings>
  <add name=
"WindowsFormsApplication1.Properties.Settings.strahovanievar30Connecti
onString"
    connectionString="server=localhost;User Id= Admin;
password=rhfvfhtyrj;Character Set= cp1251;database= strahovanievar30"
    providerName="MySql.Data.MySqlClient"/>
</connectionStrings>
```

3. Разработать интерфейс клиентского приложения.

Для того чтобы интерфейс был удобен для использования конечными пользователями, наиболее удачным вариантом будет создание Windows-приложения, состоящего из определенного количества отдельных форм, связанных с одной основной, из которой они будут вызываться.



Рис. 6. Пример формы с реализацией поиска, фильтрации, сортировки данных таблиц

Приложение должно реализовывать возможность навигации, поиска, фильтрации, сортировки данных (рис. 6), а также выполнение регламентированных запросов, экспорт данных таблиц в другие форматы и вывода на печать необходимой информации с помощью отчетов.

Для отображения наборов данных (содержимого таблиц, результатов запросов) можно использовать компонент *DataGridView* (компонент для отображения данных из таблиц, запросов или фильтров в табличном формате), или выбирать различные настройки отображения отдельных объектов набора данных (рис. 7).

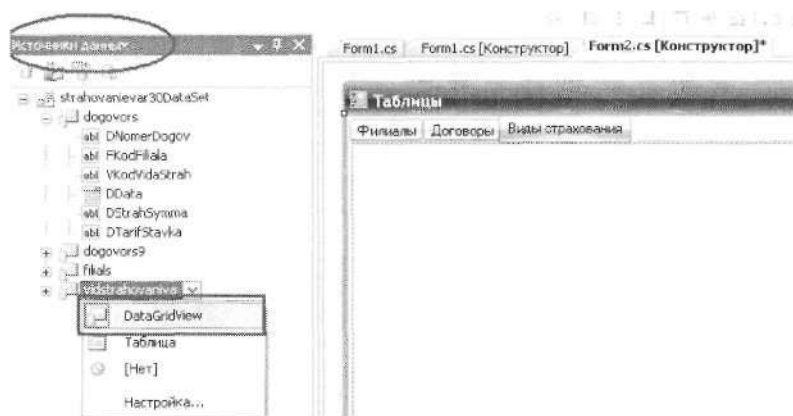


Рис. 7. Пример выбора компонентов для отображения данных таблиц

При необходимости в формах подключить в коде пространство имен для работы с базой MySQL:

```
using MySql.Data.MySqlClient;
```

Как было сказано ранее в данной статье создание интерфейса приложения рассмотрено не будет (выбор, настройку компонентов см. в [2-6]).

4. Создать отчеты Crystal Reports.NET в графическом режиме;

Пакет *Crystal Reports* для Visual Studio .NET является мощным дополнением к комплексу инструментальных средств самой среды разработки и является машиной для формирования отчетов. При помощи конструктора отчетов *Reports Designer* можно создавать и модифицировать отчеты в графическом режиме самой Visual Studio .NET, а с помощью встроенных модулей легко интегрировать отчеты в проект и получать нужную функциональность Windows- или Web-приложения.

При совместной работе технологий Crystal Reports.NET и ADO.NET доступ к данным заметно упрощается. Вместо того чтобы изобретать способы доступа к различным источникам данных, Crystal Reports .NET может просто обратиться к объекту *DataSet* (к типизированному или нетипизированному) из ADO.NET как к источнику для генерации любого заданного отчета.

Для различных СУБД возможно создание соединения с сервером посредством программного интерфейса доступа к базам данных ODBC (Open Database Connectivity). Он позволяет работать с разными источниками данных, в независимости от особенностей их взаимодействия в каждом конкретном случае.

СУБД MySQL обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC и поэтому данный механизм доступа приложений может быть использован для создания клиентского приложения. На компьютере клиента необходимо установить драйвер MyODBC и добавить новый источник данных.

Для создания отчета по технологии Crystal Reports.NET следует выполнить следующие действия:

создать новый источник данных для соединения с заданной базой данных: запустить *Администратор источников данных ODBC* из Панели управления → на вкладке *Драйверы* проверить наличие драйвера *MySQL ODBC 3.51 driver* (или выше) на компьютере пользователя (если драйвера нет в системе, то установить, например, *файл*

mysql-connector-odbc-3.51.27-win32.msi → добавить (или изменить) пользовательский DSN, выбрав нужный драйвер ODBC и задав настройки соединения (имя источника данных на выбор, ввести имя нужного сервера, зарегистрированного пользователя с паролем и выбрать имя базы данных) → проверить соединение (рис. 8).

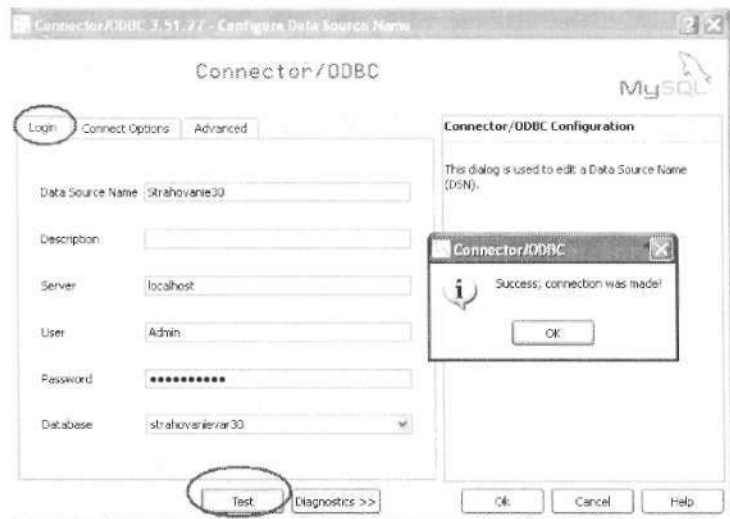


Рис. 8. Окно настройки соединения с БД

создать отчет Crystal Reports с помощью *Мастера Standard Report Creation Wizard*: добавляем из шаблонов новый элемент Crystal Report и следуем шагам мастера создания отчетов.

В окне выбора доступных источников данных для отчета выбрать создание нового соединения по технологии ODBC (RDO) (рис. 9).

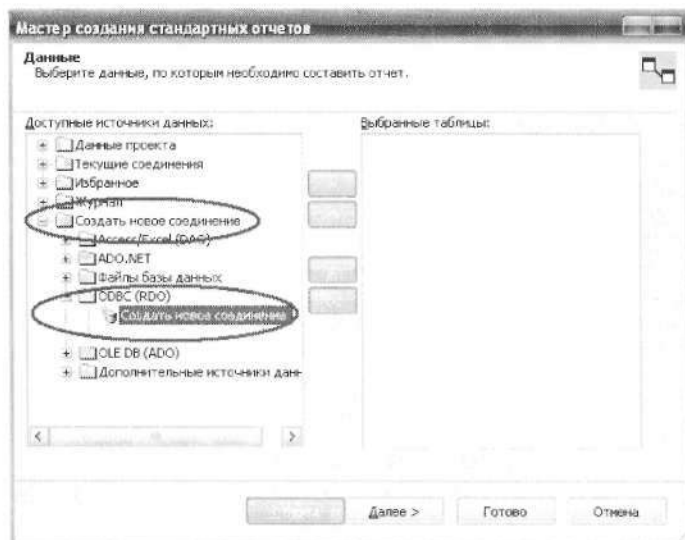
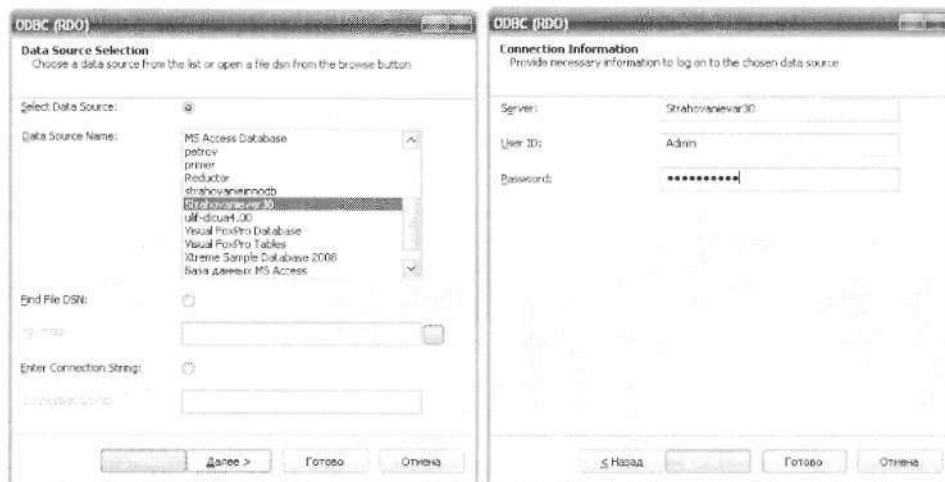


Рис. 9. Выбор данных для отчета

Далее следует выбрать нужное имя источника данных ODBC и прописать соответствующие источнику данных имя пользователя и пароль (рис. 10).



После создания отчета следует внедрить данный типизированный отчет в форму.

Выводы. Технология ADO.NET способна предоставить механизм для доступа к любому источнику данных, тем самым давая разработчику мощный механизм взаимодействия с базами данных, который реализует все потребности, возникающие при проектировании ИС. Недостатком ADO.NET является зависимость от пакета Microsoft .NET Framework, а также использование собственной логики обновления в коде, при этом собственная логика обновления занимает больше места. Однако технология ADO.NET позволяет создавать динамические запросы и запросы, выполненные на стороне сервера, что является несомненным преимуществом. Также любое .NET приложение является автономным, в том значении, что не зависит от других программ, в частности от операционной системы.

Таким образом, библиотеки платформы разработки приложений .NET Framework не имеют встроенной поддержки соединения с СУБД MySQL. Для решения данной проблемы достаточно установить драйвер провайдера .NET Framework для MySQL и в полной мере использовать возможности создания клиентского приложения базы данных для СУБД MySQL на языке программирования C# в среде Microsoft Visual Studio на платформе .NET Framework с применением механизма доступа приложений по технологии ADO.NET.

В дальнейших исследованиях будет представлен анализ программных сред, поддерживающих создание Web-интерфейсов для доступа к базам данных, построенным по технологии "клиент-сервер", в частности для СУБД MySQL.

Литература

1. Крамаренко Т. А. Рекомендации по выбору механизмов доступа приложений баз данных на C++ для СУБД MySQL в средах Code GEAR RAD Studio 2009 и Visual Studio 2005 / Т. А. Крамаренко, Ю. Л. Тихонов // Вісн. Східноукр. нац. ун-ту імені Володимира Даля. – Луганськ, 2010. – № 10 (152). – Ч. 2. – С. 107 – 114.
2. Селпа Д. Microsoft ADO.NET / Д. Селпа : Пер. с англ. – М. : Издательско-торговый дом „Русская Редакция”, 2003. – 640 с. : ил.
3. Постолиит А. В. Visual Studio .NET: разработка приложений баз данных / А. В. Постолиит. – СПб. : БХВ-Петербург, 2003. – 544 с. : ил.
4. Баженова И. Ю. Разработка распределенных приложений баз данных / И Ю. Баженова. – М. : МГУ им. М. В. Ломоносова, 2006. – 201 с.
5. Шумаков П. В. ADO.NET и создание приложений баз данных в среде Microsoft Visual Studio .NET. Руководство разработчика с примерами на C# / П. В. Шумаков. – М. : ДИАЛОГ-МИФИ, 2003. – 528 с.
6. Работа с базами данных на языке C#. Технология ADO.NET : учебное пособие / сост. О. Н. Евсеева, А. Б. Шамшев. – Ульяновск : УлГТУ, 2009. – 170 с.

Крамаренко Т.А., Козуб Г.О. Етапи створення інтерфейсу клієнтського додатка баз даних MySQL у вигляді графічного додатку з використанням технології ADO.NET

У статті розглянуто етапи створення клієнтських додатків баз даних для клієнт-серверної MySQL на мові програмування C# в середовищі Microsoft Visual Studio на платформі .NET Framework з використанням механізму доступу додатків за технологією ADO.NET.

Ключові слова: база даних, архітектура клієнт-сервер, клієнтський додаток, платформа .NET Framework, технологія ADO.NET.

Kramarenko T.A., Kozub G.A. Stages of creation of interface of client appendix of databases MySQL as graphic application with the use of technology of ADO.NET

In the article the stages of creation of client appendixes of databases are examined for client-server SUBD MySQL in programming of C# language in the environment of Microsoft Visual Studio on a platform .NET Framework with the use of mechanism of access of appendixes on technology of ADO.NET.

Keywords: database, architecture client-server, client application, platform .NET Framework, technology of ADO.NET.

Крамаренко Тетяна Анатоліївна, асистент кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка.

Козуб Галина Олександрівна, к.т.н., доцент кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка.

Онопченко Світлана Володимирівна, к.п.н., старший викладач кафедри інформаційних технологій та систем Луганського національного університету імені Тараса Шевченка.

Рецензент: **Белодедов Виктор Александрович**, д.т.н., проф. ВЛУ.

Стаття подана 13.06.2012

УДК. 004.056.

Лахно В.А.

УПРАВЛЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬЮ НА ТРАНСПОРТЕ В УСЛОВИЯХ ДЕСТРУКТИВНОГО ВОЗДЕЙСТВИЯ НА ИНФОРМАЦИОННЫЕ СИСТЕМЫ

В статье предложена модель системы управления информационной безопасностью на транспорте, обеспечивающая оценку уровня риска ее нарушения и поддержку принятия решений относительно противодействия несанкционированного доступа к информации. Ист. 11.

Ключевые слова: защита информации, информационная безопасность, информационный риск, принятие решения.

Постановка проблемы. Защита информации, информационная безопасность (ИБ) начинают играть одну из ключевых ролей в обеспечении важных, прежде всего, экономических интересов в хозяйственном комплексе страны и, в частности, тех отраслях, которые связаны с транспортной и информационной инфраструктурой. Тот факт, что Украина по своему транзитному потенциалу занимает одно из первых мест в Европе, определяет особую значимость разработки и проведения эффективной транспортной политики. Ведущую роль в ней должны играть ускоренное развитие транспортной и информационной инфраструктуры, создание соответственно международным стандартам национальной сети международных транспортных коридоров, ее интегрирование в транспортные системы Европы и Азии. Динамично нарастающая информатизация транспортной отрасли рассматривается как необходимая техническая и информационная основа совершенствования технологического процесса работы транспортной отрасли. За последние десятилетия создана развитая информационная инфраструктура и накоплены огромные массивы информации, носящей в том числе, конфиденциальный характер. Повысились требования к безопасности внедряемых на транспорте информационных технологий.

**ВІСНИК
СХІДНОУКРАЇНСЬКОГО НАЦІОНАЛЬНОГО
УНІВЕРСИТЕТУ
імені ВОЛОДИМИРА ДАЛЯ**

№ 8 (179) 2012

Науковий журнал

Відповідальний за випуск Стативка Ю.І.

Технічний редактор
Оригінал-макет

Т.М. Дроговоз
О.В. Могильна

Підписано до друку 18.06.2012 р.
Формат 70 x 108 1 /16. Папір офсетний. Гарнітура Times New Roman.
Друк офсетний. Умов. друк. арк. 28.0. Обл. друк. арк. 29.1.
Наклад 300 прим. Вид. № 2768. Замовлення № 73 . Ціна вільна.

Видавництво
Східноукраїнського національного університету
імені Володимира Даля

Свідоцтво про реєстрацію: серія ДК № 1620 від 18.12.03 р.

Адреса видавництва: 91034, м. Луганськ, кв. Молодіжний, 20 а,
Телефон (0642) 41-34-12. Факс (0642) 41-31-60.
E-mail: uni@snu.edu.ua