

№ 11 Создание WPF приложения управления базой данных

Задание

- 1) Установите и настройте MS SQL Server.
- 2) Создайте новую базу данных, а также необходимую инфраструктуру (таблицы, связи между таблицами, триггеры, хранимые процедуры и т.д.) на основе вариантов в лаб. 5 (желательно 3-4 таблицы, минимум должно **быть 2 таблицы**). Одно из полей БД должно быть графическая информация (фото, логотип, карта проезда и т.п.).
- 3) создать приложение выполняющего CRUD функции баз данных на основе ADO.NET.
- 4) Обеспечить следующий функционал:
 - a. добавление, редактирование, удаление и просмотр записей из базы данных;
 - b. сортировка;
 - c. Выполнение запросов, запросов с параметрами и хранимых процедур.
- 5) При работе с базой данных используйте транзакции.
- 6) Отображайте информацию в виде таблицы (можно DataGrid). Должны быть кнопки навигации (→ ← и т.п.). Обязательно использовать привязки данных к ЭУ.
- 7) Сохраните строку подключения и другие параметры работы приложения в конфигурационном файле (например, App.config).
- 8) База данных может быть любая реляционная (MS SQL, MySQL, SQLite и т.п., **Access – нельзя**).
- 9) В случае если после запуска приложения не удастся найти БД по указанному пути, она должна создаваться во время работы программы (должны быть скрипты создания).

P.S. По согласованию с преподавателем, для разработки можно взять базу данных из курсового проекта (если такая имеется)

Вопросы

1. Что такое Binding в WPF? Какие атрибуты мы можем указывать и для чего?
2. Что такое ADO.NET? Какие существуют режимы работы?
3. Объясните назначение класса SqlConnection.
4. Объясните назначение класса SqlDataAdapter
5. Объясните назначение класса SqlException. Чем данный класс отличается от класса-родителя Exception?
6. Объясните назначение класса SqlCommand.
7. В чем разница между методами ExecuteNonQuery, ExecuteReader и ExecuteScalar?
8. Объясните назначение класса SqlDataReader.
9. Объясните назначение класса SqlParameter.
10. Объясните назначение класса SqlTransaction.
11. Объясните назначение классов DataSet, DataTable, DataColumn, DataRow, DataRelation.

Краткие теоретические сведения

Обзор объектов ADO.NET

Источник данных DataSet

Основным объектом ADO является источник данных, представленный объектом DataSet. DataSet состоит из объектов типа DataTable и объектов DataRelation. В коде к ним можно обращаться как к свойствам объекта DataSet, то есть, используя точечную нотацию. Свойство Tables возвращает объект типа DataTableCollection, который содержит все объекты DataTable используемой базы данных.

Таблицы и поля (объекты DataTable и DataColumn)

Объекты DataTable используются для представления одной из таблиц базы данных в DataSet. В свою очередь, DataTable составляется из объектов DataColumn.

DataColumn - это блок для создания схемы DataTable. Каждый объект DataColumn имеет свойство DataType, которое определяет тип данных, содержащихся в каждом объекте DataColumn. Например, можно ограничить тип данных до целых, строковых и десятичных чисел. Поскольку данные, содержащиеся в DataTable, обычно переносятся обратно в исходный источник данных, необходимо согласовывать тип данных с источником.

Объекты DataRelation

Объект DataSet имеет также свойство Relations, возвращающее коллекцию DataRelationCollection, которая в свою очередь состоит из объектов DataRelation. Каждый объект DataRelation выражает отношение между двумя таблицами (сами таблицы связаны по какому-либо полю (столбцу)). Следовательно, эта связь осуществляется через объект DataColumn.

Строки (объект DataRow)

Коллекция Rows объекта DataTable возвращает набор строк (записей) заданной таблицы. Эта коллекция используется для изучения результатов запроса к базе данных. Мы можем обращаться к записям таблицы как к элементам простого массива.

DataAdapter

DataSet - это специализированный объект, содержащий образ базы данных. Для осуществления взаимодействия между DataSet и собственно источником данных используется объект типа DataAdapter. Само название этого объекта - адаптер, преобразователь, - указывает на его природу. DataAdapter содержит метод Fill() для обновления данных из базы и заполнения DataSet.

Объекты DBConnection и DBCommand

Объект DBConnection осуществляет связь с источником данных. Эта связь может быть одновременно использована несколькими командными объектами. Объект DBCommand позволяет послать базе данных команду (как правило, команду SQL или хранимую процедуру). Объекты DBConnection и DBCommand иногда создаются неявно в момент создания объекта DataSet, но их также можно создавать явным образом.

Используйте примеры:

<https://msdn.microsoft.com/ru-ru/library/h0y4a0f6.aspx>