

Лабораторная работа №9

Изучение классов автономного доступа к БД из приложения C#

1Цель работы

- 1.1 Изучить свойства и методы компонентов SqlDataAdapter, SqlCommandBuilder, DataTable, научиться их применять и настраивать;
- 1.2 Научиться создавать приложение C# для организации доступа к БД.

2Литература

- 21 Фленов, М.Е. Библия C#. 3 изд. – Санкт-Петербург: БХВ-Петербург, 2016. – URL: <https://ibooks.ru/bookshelf/353561/reading>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.16.

3Подготовка к работе

- 3.1 Повторить теоретический материал (см.п.2).
- 3.2 Изучить описание лабораторной работы.

4Основное оборудование

- 4.1 Персональный компьютер.

5Задание

Для выполнения потребуется статический класс DAL (DataAccessLayer) из ЛР №8 с методом или свойством для получения строки подключения.

5.1 Создание метода для получения данных из таблицы БД

5.1.1 Добавить в класс DAL открытый метод, принимающий строку с названием таблицы БД, выполняющий команду на выборку всех данных из таблицы БД и возвращающий результат в виде DataTable.

Для заполнения использовать SqlDataAdapter.

При подключении к БД должен использоваться метод (или свойство на чтение), возвращающий строку подключения к БД.

5.1.2 Добавить в приложение новую форму и реализовать переход к ней. Проверить работу созданного метода, вызвав его при нажатии кнопки «Показать» на форме. Название таблицы должно вводиться пользователем в поле ввода, результат выполнения команды должен отображаться на экране в DataGridView.

5.2 Настройка родительской формы для отображения данных таблиц БД

5.2.1 Добавить в приложение форму для отображения данных и выполнить следующее:

- разместить на форме DataGridView (привязать его к краям формы), кнопки «Сохранить» и «Обновить»,
- добавить в класс формы защищенное строковое поле для хранения названия таблицы,
- добавить в класс формы поле типа DataTable для хранения содержимого таблицы.

5.2.2 На загрузку формы и нажатие кнопки «Обновить» реализовать заполнение DataTable, используя метод из п.5.1.1. Присвоить свойству TableName у DataTable значение строкового поля. Указать DataTable как источник данных для DataGridView.

5.2.3 Реализовать перехват исключений при загрузке и информирование пользователя о возникших ошибках.

5.3 Создание дочерних форм для отображения данных таблиц БД

5.3.1 Добавить в приложение формы «Товары», «Производители», «Заказчики» – наследники формы из п.5.2. В конструкторах дочерних форм изменить:

- значение строкового поля на название требуемой таблицы,
- заголовок формы на соответствующий ее содержимому.

5.3.2 Добавить в приложение форму с набором кнопок для перехода к формам из п.5.1.2 и п.5.3.1. Изменить приложение так, чтобы эта форма открывалась первой при запуске приложения.

5.4 Создание метода для сохранения данных таблицы БД

5.4.1 Добавить в класс DAL открытый метод, принимающий объект DataTable и строку с названием таблицы БД (по умолчанию — null) и выполняющий обновление данных указанной таблицы БД.

Объект DataTable должен передаваться по ссылке (ref).

Если название таблицы не передано, брать его из DataTable (свойство TableName).

Для обновления использовать SqlDataAdapter и SqlCommandBuilder.

При подключении к БД должен использоваться метод (или свойство на чтение), возвращающий строку подключения к БД.

5.4.2 Проверить работу созданного метода, вызвав его в форме из п.5.1.2, отображающей данные таблицы БД. Метод должен вызываться при нажатии на кнопку «Сохранить».

5.5 Настройка родительской формы для сохранения данных таблиц БД

5.5.1 На нажатие кнопки «Сохранить» реализовать сохранение данных поля DataTable, используя метод из п.5.4.1.

5.5.2 Реализовать перехват исключений при сохранении и информирование пользователя о возникших ошибках.

5.5.3 Проверить работу сохранения в дочерних формах из п.5.3.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать на C# оконное приложение.

6.2 Выполнить все задания из п.5. При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать. Исключения должны обрабатываться в основной программе. При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8Контрольные вопросы

- 8.1 Для чего применяется компонент SqlDataAdapter?
- 8.2 Для чего применяется компонент SqlCommandBuilder?
- 8.3 Как получить данные из БД, используя SqlDataAdapter?
- 8.4 Как обновить данные в БД, используя SqlDataAdapter?
- 8.5 Как связать SqlCommandBuilder и SqlDataAdapter?