Лабораторная работа №7

Создание программы для работы с БД

Цель: изучить основные возможности программного взаимодействия приложений .NET с базами данных под управлением СУБД Access.

Для получения **теоретических сведений** настоятельно рекомендуется при домашней подготовке изучить материалы по тематике лабораторной работы, представленные в открытых источниках.

Далее следует краткое описание примера создания в среде Visual Studio 2015 элементарного приложения для работы с БД в СУБД MS Access.

Пример создания приложения для работы с БД в Visual Studio 2015

Для создания приложения на языке C# в среде Microsoft Visual Studio 2015 необходимо при создании проекта выбрать язык Visual C#, в качестве шаблона создаваемого приложения – выбрать **Windows Forms Application**, после чего указать имя нового проекта в поле **Имя (Name)** (см. рис 1.).

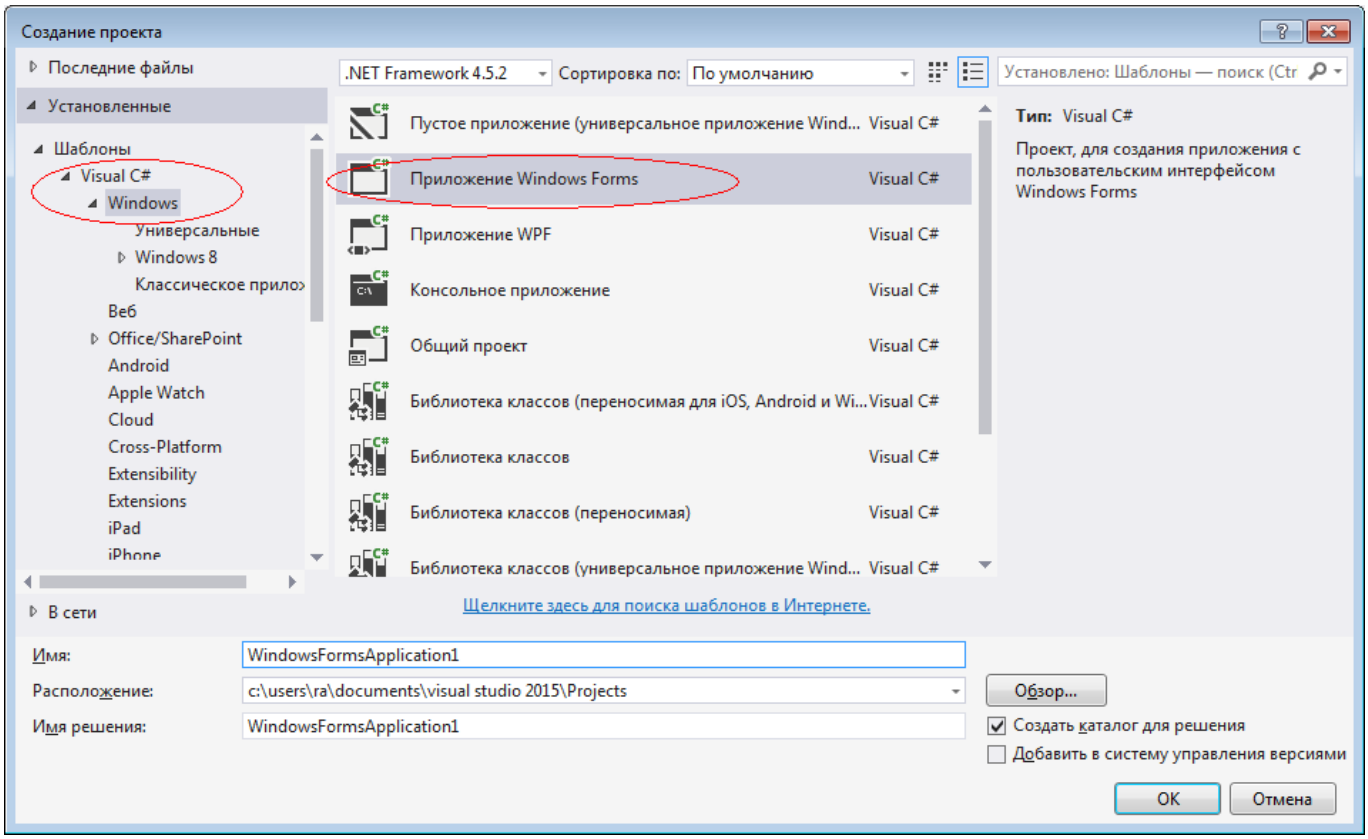

Рисунок 1 – Выбор шаблона приложения

Далее, после формирования средой файлов проекта, можно открыть необходимую БД прямо в Visual Studio для этого, в окне **«Обозреватель серверов»** (**Server Explorer**) установите соединение с БД:

щелчок правой кнопкой мыши по **Подключения данных** (**Data Connections**) -> **Добавить подключение... (Add Connection...)** (см. рис. 2).

Далее, в окне **«Выбор источника данных»** укажите источник данных (см. рис. 3)**,** далее в окне **«Добавить подключение»** (**Add Connection**) указывается файл, в котором хранится БД (см. рис. 4).

После чего, структура БД доступна для просмотра в окне **«Обозреватель серверов»** (**Server Explorer**) (см. рис. 5).

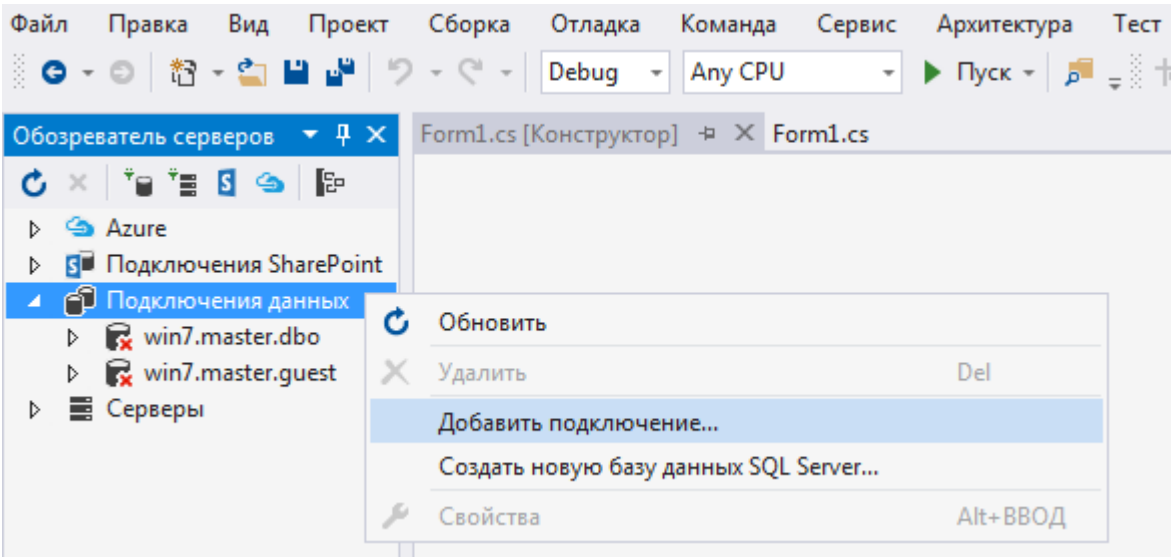

Рисунок 2. Подключение данных

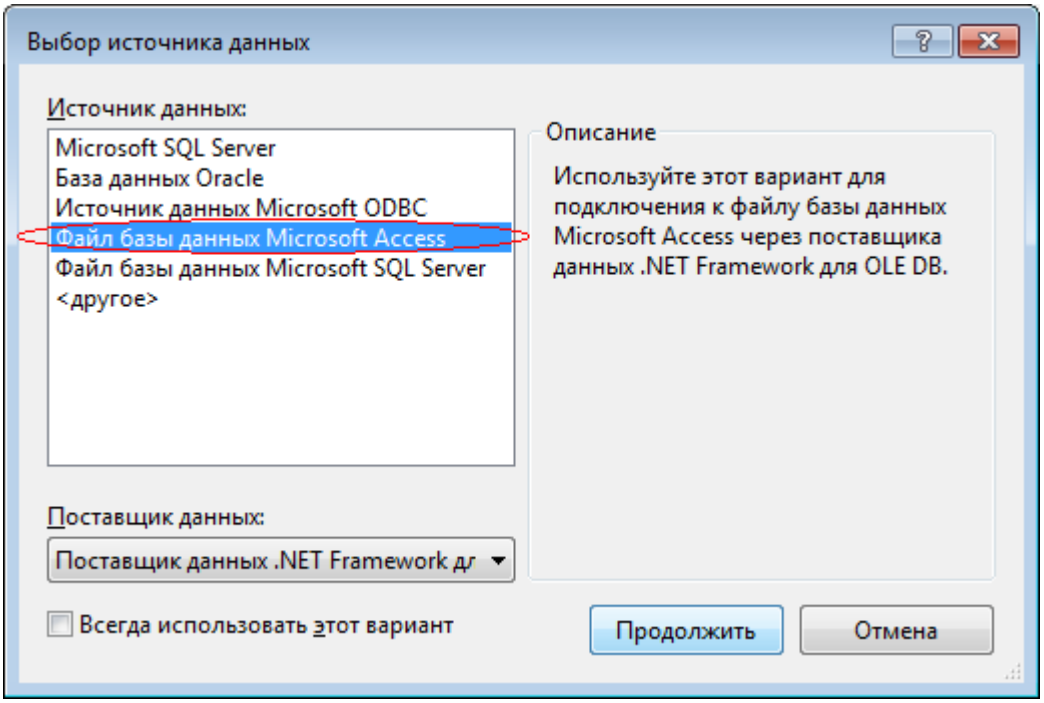

Рисунок 3. Выбор источника данных

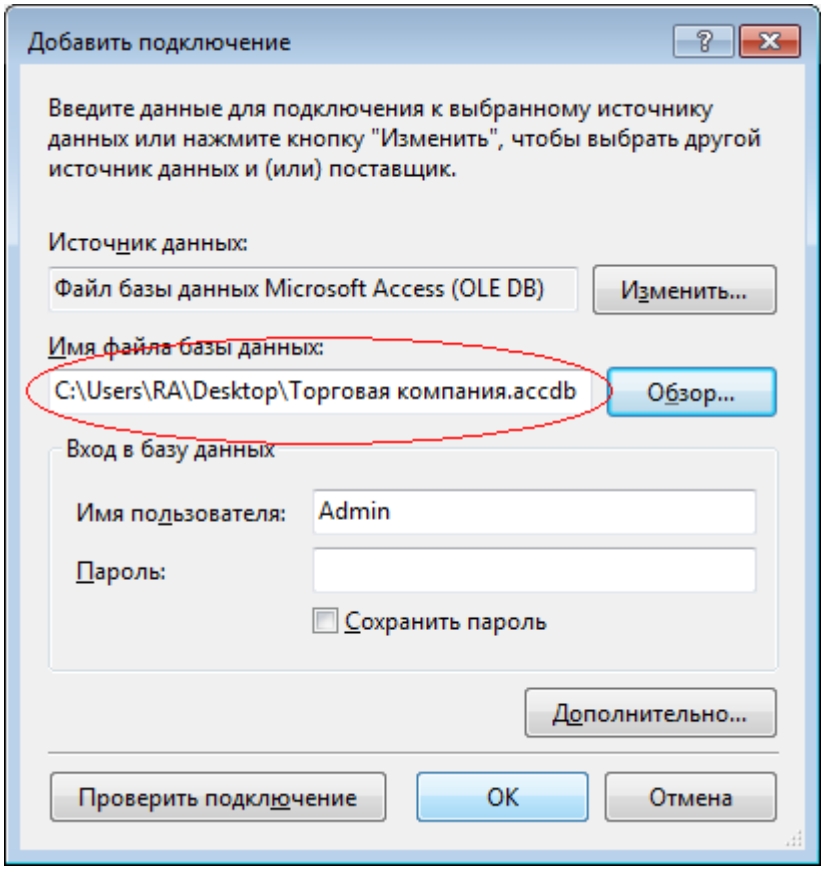

Рисунок 4. Добавление подключения

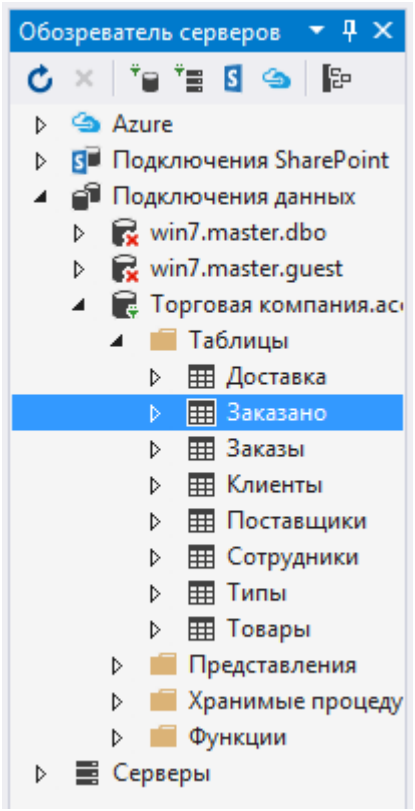

Рисунок 5. Вид окна «Обозреватель серверов» (Server Explorer) после соединения с БД

Далее, **приведем сам пример** создания элементарного приложения, взаимодействующего с БД. Приложение будет при нажатии на одну из кнопок выводить список сотрудников из учебной БД «Торговая компания», при нажатии на другую – добавлять в БД нового сотрудника с заданными именем и фамилией, а при нажатии на третью – удалять из БД сотрудника с заданной фамилией. Четвертая кнопка будет выполнять сохраненный запрос.

Вначале добавим на форму главного окна приложения несколько визуальных компонентов (компоненты находятся в окне «Панель элементов» (Toolbox) и добавляются путем простого перетаскивания на форму): три компонента типа Button (кнопки), два компонента типа TextBox (поля ввода) и один – типа ListBox (список) (см. рис. 6).

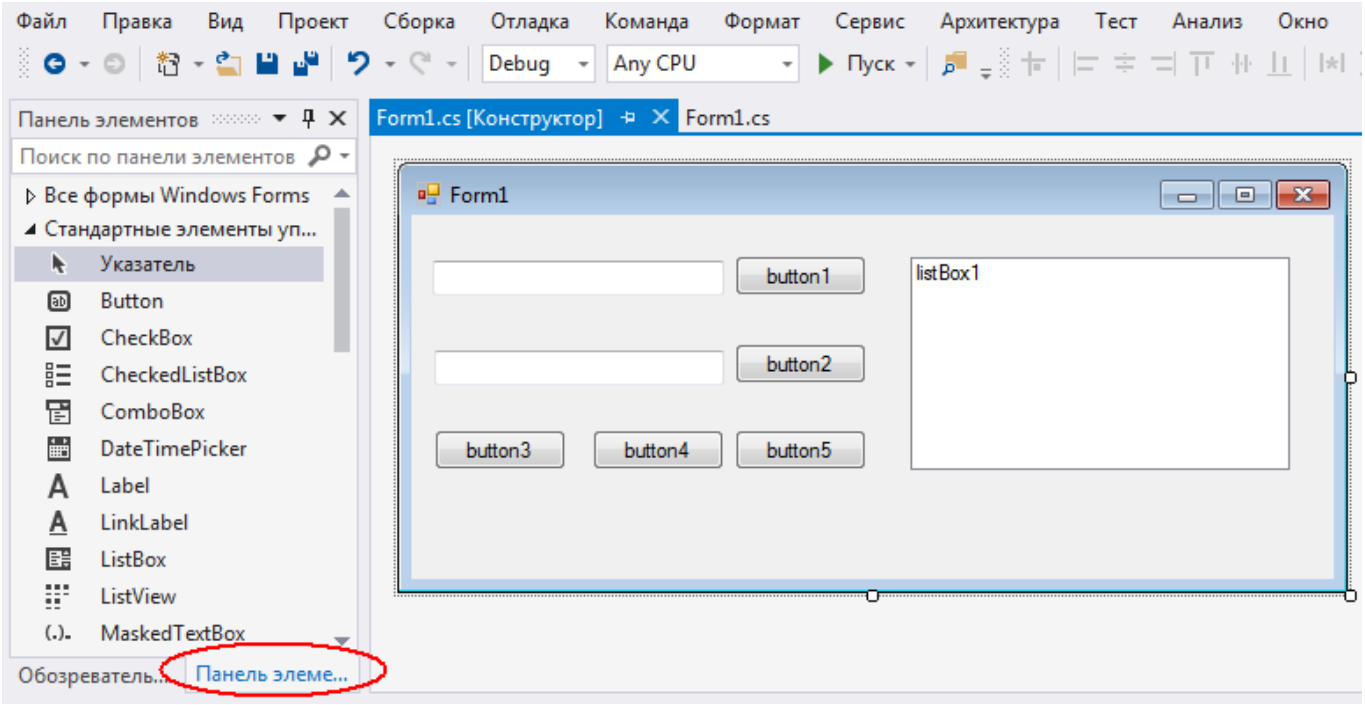

Рисунок 6. Добавление компонентов на форму

Далее необходимо перейти к описанию класса главной формы (для этого можно просто дважды щелкнуть по самой форме).

В коде, для взаимодействия программы с БД посредствам СУБД MS Access необходимо:

- подключить объекты пространства имен **`System.Data.OleDb`**, прописав строку:

```
using System.Data.OleDb;
```

после других строк, начинающихся со слова **`using`**;

- создать объект типа **`OleDbConnection`** для установки соединения с БД, в конструктор этого объекта в качестве параметра нужно передать строку подключения, посмотреть ее для конкретной БД можно в окне **Свойства (Properties)** (см. рис. 7);

- создать объект типа **`OleDbCommand`** для отправки СУБД Access SQL-запросов.

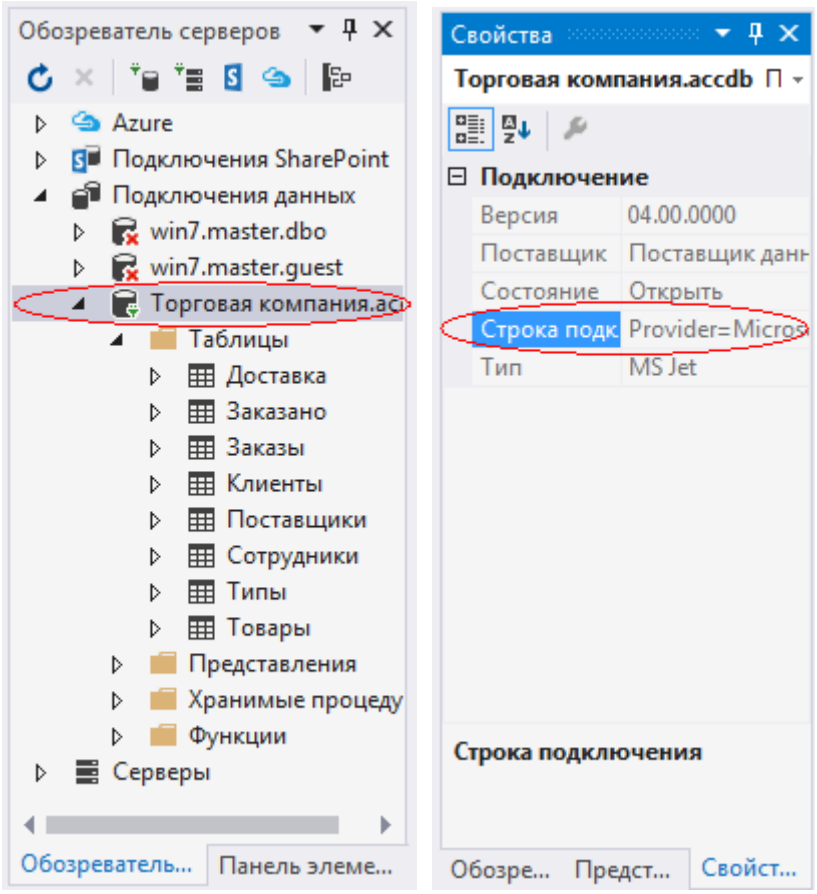

Рисунок 7. Просмотр строки подключения

На первом этапе по нажатию кнопки button1 будем выводить фамилии из таблицы сотрудники:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        OleDbConnection cn = new OleDbConnection(
            @"Provider=Microsoft.ACE.OLEDB.12.0;"+
            @"Data Source=""C:\Users\RA\Desktop\Торговая компания.accdb"""
        );

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
```

```
        // метод, возвращающий список фамилий сотрудников из БД
        private List<String> Get_Employees_Families()
        {
            List<String> res = new List<String>();
            cn.Open();   // установка соединения
            try
            {
                OleDbCommand cmd = new OleDbCommand();
                // установка связи между объектом отправки SQL-запросов и
                // соединением
                cmd.Connection = cn;
                // создание запроса на выбору с параметром (@Templ)
                cmd.CommandText =
                  "SELECT * FROM Сотрудники WHERE Фамилия LIKE @Templ";
                // установка значения параметра
                cmd.Parameters.AddWithValue("@Templ", "%в");

                // попытка выполнить запрос,
                // доступ к его результату будет осуществляться
                // при помощи объекта rd
                OleDbDataReader rd = cmd.ExecuteReader();

                // если запрос вернул результат
                if (rd.HasRows)
                {
                    // получаем строки одну за другой, и для каждой строки...
                    while (rd.Read())
                    {
                        // ... добавляем в res содержимое столбца «Фамилия»
                        res.Add(rd.GetValue(rd.GetOrdinal("Фамилия"))
                           .ToString());
                    }
                }
                return res;
            }
            finally
            {
                cn.Close();   // закрытие соединения с БД
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
            // переносим список фамилий в listBox1
            foreach (String i in Get_Employees_Families())
                listBox1.Items.Add(i);
        }
    }
```

В том случае, когда название(я) столбца(ов) неизвестны, можно получить к нему(им) доступ по номеру:

```
//...
while (rd.Read())
{
    res.Add(rd[1].ToString());
}
```

На следующем этапе добавим возможность при нажатии кнопки button2 добавлять в таблицу «Сотрудники» нового сотрудника, чья фамилия указана в поле ввода textBox1, а имя – в textBox2:

```
private void Add_new_Employee(String family, String name)
{
    cn.Open();
    try
    {
```

```
            OleDbCommand cmd = new OleDbCommand();
            cmd.Connection = cn;
            cmd.CommandText =
                "INSERT INTO Сотрудники(Фамилия, Имя) VALUES (@Family, @Name)";

            cmd.Parameters.AddWithValue("@Family", family);
            cmd.Parameters.AddWithValue("@Name", name);
            cmd.ExecuteNonQuery();
        }
        finally
        {
            cn.Close();
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Add_new_Employee(textBox1.Text, textBox2.Text);
    }
```

А затем - возможность при нажатии кнопки button3 удалять из таблицы «Сотрудники» сотрудников, чья фамилия совпадает с данными, введенными в textBox1:

```
    private void Delete_Employee(String family)
    {
        cn.Open();
        try
        {
            OleDbCommand cmd = new OleDbCommand();
            cmd.Connection = cn;
            cmd.CommandText = "DELETE FROM Сотрудники WHERE Фамилия=@Family";

            cmd.Parameters.AddWithValue("@Family", family);
            cmd.ExecuteNonQuery();
         }
         finally
         {
            cn.Close();
         }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Delete_Employee(textBox1.Text);
    }
```

Программный вызов пользовательских запросов к БД

Рассмотрим некоторые особенности устройства СУБД MS Access.

1. Ядро СУБД MS Access построено на основе ядра СУБД MS SQL Server, вследствие этого, несмотря на ограничения, имеющиеся в MS Access на уровне пользователя, при программном вызове SQL-запросов, наложенные ограничения отсутствуют (хотя отличия имеются всё равно).

2. Пользовательские объекты, сохраняемые в MS Access как «Запросы», на самом деле являются либо **представлениям** (запросы на выборку без параметров), либо **функциями** (запросы на выборку с параметрами), либо **хранимыми процедурами** (запросы на добавление, обновление и удаление).

Выборка из представлений происходит как из таблиц (через оператор SELECT).

Вызов функций также происходит посредствам оператора SELECT, однако перед этим необходимо задать значения параметров функции (заметим, что в MS Access не поддерживает всех возможностей синтаксиса Transact-SQL, в результате необходимо использовать в коде программы функцию для передачи параметров в запрос), например:

создадим в учебной БД «Торговая компания» запрос «Выборка1» с параметром **xxx**:

```
SELECT * FROM Клиенты WHERE Должность=[xxx]
```

Тогда, программно получить результат данного запроса можно следующим образом:

```
// ...
// Делаем выборку, передавая в функцию Выборка1
// в качестве параметра xxx значение Продавец
cmd.Parameters.AddWithValue("xxx", "Продавец");
cmd.CommandText = "SELECT * FROM Выборка1";
// ...
```

Вызов хранимых процедур осуществляется посредствам стандартного синтаксиса SQL, например:

создадим в учебной БД «Торговая компания» запрос «Выборка2» с параметрами **x1, x2** и **x3**:

```
INSERT INTO Доставка(КодДоставки, Название, Телефон) VALUES ([x1],[x2],[x3]);
```

Тогда, программно получить результат данного запроса можно следующим образом:

```
private void Procedure_Call_Example()
{
    cn.Open();
    try
    {
        OleDbCommand cmd = new OleDbCommand();
        cmd.Connection = cn;
        cmd.CommandText = "EXEC Выборка2 8, 'qwerty', '(495) 123-5678'";
        cmd.ExecuteNonQuery();
    }
    finally
    {
        cn.Close();
    }
}

private void button4_Click(object sender, EventArgs e)
{
    Procedure_Call_Example();
}
```

3. Названия и структуры таблиц, в которых хранится информация о БД в MS Access, также отличаются от тех, что предписывает стандарт SQL. Например, в MS Access информация обо всех объектах БД хранится в таблице **MSysObjects**.

Для получения программного доступа к таблице MSysObjects и прочим служебным таблицам MS Access, необходимо в строке подключения прописать разрешение на доступ к служебным таблицам и путь к локальному системному файлу Access - **System.mdw**. Обычно он располагается в папке:

C:\Documents and Settings\<**username**>\Application Data\Microsoft\Access\

где <**username**> - имя пользователя.

В результате, задание строки подключения может выглядеть, например, так:

```
OleDbConnection cn = new OleDbConnection(
    @"Provider=Microsoft.ACE.OLEDB.12.0;"+
    @"Data Source=""C:\Users\RA\Desktop\Торговая компания.accdb"";"+
    @"Jet OLEDB:Create System Database=true;" +       // разрешение на доступ
    @"Jet OLEDB:System database=C:\Documents and Settings\RA\" +
    @"Application Data\Microsoft\Access\System.mdw"
);
```

Лабораторное задание

Задание 1. Напишите программу для взаимодействия с БД, разработанной на предыдущих лабораторных работах.

Требования к программе:

- графический интерфейс;
- возможность отображения, добавления, изменения и удаления данных для каждой таблицы;
- корректное поведение при возникновении ошибок на стороне СУБД;
- корректное поведение при некорректных действиях пользователя.

Задание 2. Сделав запрос к таблице **MSysObjects**, изучите состав БД, разработанной на предыдущих Л.р:

- попытайтесь определить назначение столбцов этой таблицы;
- попытайтесь определить смысл значений столбца **Type**;
- на основании результата запроса к **MSysObjects** определите набор прочих служебных таблиц MS Access, изучите их содержимое и попытайтесь определить назначение.

В случае если БД разрабатывалась не в Access, самостоятельно изучите способы получения метаданных в используемой Вами СУБД.

Результаты занесите в отчет.

Задание 3. Дополните программу возможностью вызова (и получения результатов) запросов, добавленных в БД на Л.р.№№4,5,6: не менее **3 запросов на выборку без параметров**, не менее **3 запросов на выборку с параметрами**, и, минимум, по одному запросу **на добавление, обновление и удаление**, предусмотрев в интерфейсе программы соответствующие изменения. Для определения наличия в БД соответствующих запросов использовать выборку из таблицы **MSysObjects**.

В случае если БД разрабатывалась не в Access, самостоятельно изучите способы выполнения запросов, сохраненных в БД (к примеру, в виде представлений, функций, хранимых процедур).

Требования к сдаче работы

1. Продемонстрировать работу **программы и программный код**.
2. Продемонстрировать отчет о выполнении **Задания 2**.
3. Ответить на контрольные вопросы и выполнить дополнительные задания, демонстрирующие полученные навыки.

Дополнительные ссылки

1. Работа с базой данных SQL Server с использованием C#:
   https://o7planning.org/ru/10515/working-with-sql-server-database-using-csharp
2. Просто SQL select в C#?
   https://coderoad.ru/1504895/Просто-SQL-select-в-C
3. Руководство по ADO.NET и работе с базами данных
   https://metanit.com/sharp/adonet/