MS Access и C# – **работаем с базой данных из программы Windows Forms**

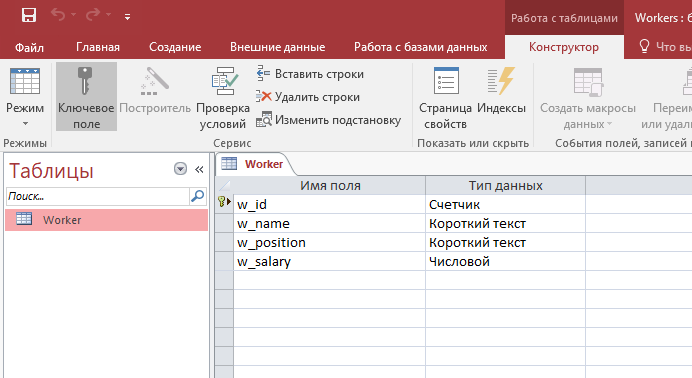
Рассмотрим выполнение запросов SELECT, INSERT, UPDATE и DELETE к базе данных Microsoft Access из программы на языке C#, на примере проекта Windows Forms в Visual Studio. В конце статьи вы найдете исходники, а также видеоурок по данной теме.

Проектирование базы данных MS Access

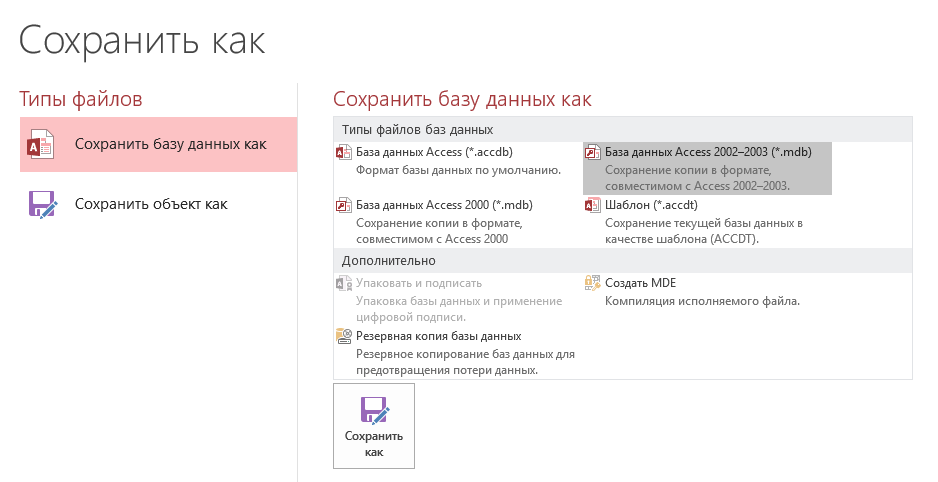
Создадим в Microsoft Access базу данных Workers (рабочие) с одной таблицей Worker (рабочий).

Таблица содержит следующие столбцы:

* w\_id (идентификатор записи) – тип данных Счетчик;
* w\_name (имя) – тип данных Короткий текст;
* w\_position (должность) – тип данных Короткий текст;
* w\_salary (зарплата) – тип данных Числовой.

[](https://vscode.ru/wp-content/uploads/2018/03/Shema-bazyi-dannyih-MS-Access.png)

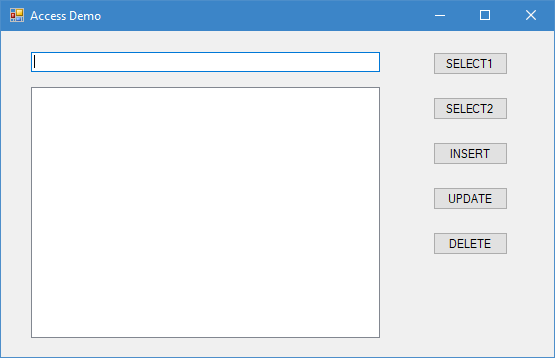
Для корректной работы с базой данных MS Access с помощью языка C#, ее необходимо сохранить в формате с расширением **.mdb**

[](https://vscode.ru/wp-content/uploads/2018/03/Sohranenie-BD-Access-v-nuzhnom-formate-dlya-C.png)

Создание проекта Windows Forms в Visual Studio на языке C#

Создадим проект Windows Forms на C# и поместим на форму следующие компоненты:

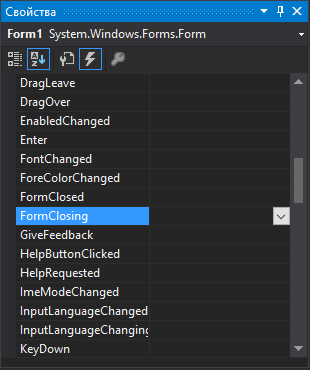
TextBox, ListBox и пять кнопок Button.

[](https://vscode.ru/wp-content/uploads/2018/03/Raspolozhenie-e%60lementov-upravleniya-na-forme.png)

На макете формы в Visual Studio выполним двойные щелчки левой кнопкой мыши на всех кнопках, создав тем самым обработчики событий нажатия для каждой кнопки.

|  |  |
| --- | --- |
| 1  2  3  4  5 | // обработчик события нажатия кнопки SELECT1  private void selectButton1\_Click(object sender, EventArgs e)  {    } |

Также в окне свойств формы с помощью двойного щелчка мыши создадим обработчик события FormClosing, выполняющегося при закрытии формы.

[](https://vscode.ru/wp-content/uploads/2018/03/Obrabotchik-sobyitiya-zakryitiya-formyi.png)

|  |  |
| --- | --- |
| 1  2  3  4  5 | // обработчик события закрытия формы  private void Form1\_FormClosing(object sender, FormClosingEventArgs e)  {    } |

Файл с БД поместим в каталог \bin\Debug в папке с проектом.

Написание кода на C# для выполнения запросов к MS Access

Добавим в класс формы строковое поле string – строку подключения к БД. В строке подключения в значении параметра Data Source указывается путь к файлу с базой данных.

|  |  |
| --- | --- |
| 1  2  3  4  5 | // строка подключения к MS Access  // вариант 1  public static string connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=Workers.mdb;";  // вариант 2  //public static string connectString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=Workers.mdb;"; |

Есть два варианта строки с различными провайдерами подключения. Если с первой строкой программа не будет работать – закомментируйте ее, а вторую раскомментируйте.

Далее добавим поле класса типа OleDbConnection. Экземпляр данного класса понадобится для установления соединения с базой данных MS Access.

|  |  |
| --- | --- |
| 1  2 | // поле - ссылка на экземпляр класса OleDbConnection для соединения с БД  private OleDbConnection myConnection; |

В конструкторе класса создадим экземпляр OleDbConnection и установим соединение с БД. Т.е. при запуске программы будет устанавливаться соединение с Access.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | // конструктор класса формы  public Form1()  {      InitializeComponent();        // создаем экземпляр класса OleDbConnection      myConnection = new OleDbConnection(connectString);        // открываем соединение с БД      myConnection.Open();  } |

В обработчике события закрытия формы добавим код, закрывающий соединение с базой данных. Таким образом при выходе из программы соединение с БД будет закрываться.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // обработчик события закрытия формы  private void Form1\_FormClosing(object sender, FormClosingEventArgs e)  {      // заркываем соединение с БД      myConnection.Close();  } |

Напишем в обработчик события нажатия кнопки SELECT1 код, выполняющий запрос выборки данных одного столбца одной строки. Вывод данных производится в TextBox.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // обработчик события нажатия кнопки SELECT1  private void selectButton1\_Click(object sender, EventArgs e)  {      // текст запроса      string query = "SELECT w\_name FROM Worker WHERE w\_id = 1";        // создаем объект OleDbCommand для выполнения запроса к БД MS Access      OleDbCommand command = new OleDbCommand(query, myConnection);        // выполняем запрос и выводим результат в textBox1      textBox1.Text = command.ExecuteScalar().ToString();  } |

При нажатии кнопки SELECT2 будет выполняться запрос на получение многострочных данных из нескольких столбцов (по сути таблицы). Данные выводятся в ListBox.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | // обработчик события нажатия кнопки SELECT2  private void selectButton2\_Click(object sender, EventArgs e)  {      // текст запроса      string query = "SELECT w\_name, w\_position, w\_salary FROM Worker ORDER BY w\_salary";        // создаем объект OleDbCommand для выполнения запроса к БД MS Access      OleDbCommand command = new OleDbCommand(query, myConnection);        // получаем объект OleDbDataReader для чтения табличного результата запроса SELECT      OleDbDataReader reader = command.ExecuteReader();        // очищаем listBox1      listBox1.Items.Clear();        // в цикле построчно читаем ответ от БД      while(reader.Read())      {          // выводим данные столбцов текущей строки в listBox1          listBox1.Items.Add(reader[0].ToString() + " " + reader[1].ToString() + " " + reader[2].ToString() + " ");      }        // закрываем OleDbDataReader      reader.Close();  } |

Далее запрос вставки данных (INSERT) в MS Access на C#:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // обработчик события нажатия кнопки INSERT  private void insertButton\_Click(object sender, EventArgs e)  {      // текст запроса      string query = "INSERT INTO Worker (w\_name, w\_position, w\_salary) VALUES ('Михаил', 'Водитель', 20000)";        // создаем объект OleDbCommand для выполнения запроса к БД MS Access      OleDbCommand command = new OleDbCommand(query, myConnection);        // выполняем запрос к MS Access      command.ExecuteNonQuery();  } |

Метод ExecuteNonQuery() класса OleDbCommand выполняет запрос и возвращает целое число типа int – количество строк затронутых выполняемым запросом. Данный метод подходит для запросов вставки INSERT, обновления UPDATE и удаления DELETE. Т.е. для тех, которые не возвращают данные.

Запрос обновления данных UPDATE:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // обработчик события нажатия кнопки UPDATE  private void updateButton\_Click(object sender, EventArgs e)  {      // текст запроса      string query = "UPDATE Worker SET w\_salary = 123456 WHERE w\_id = 3";        // создаем объект OleDbCommand для выполнения запроса к БД MS Access      OleDbCommand command = new OleDbCommand(query, myConnection);        // выполняем запрос к MS Access      command.ExecuteNonQuery();  } |

Пример SQL-запроса удаления данных (DELETE) из БД Access с помощью языка C#:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // обработчик события нажатия кнопки DELETE  private void deleteButton\_Click(object sender, EventArgs e)  {      // текст запроса      string query = "DELETE FROM Worker WHERE w\_id < 3";        // создаем объект OleDbCommand для выполнения запроса к БД MS Access      OleDbCommand command = new OleDbCommand(query, myConnection);        // выполняем запрос к MS Access      command.ExecuteNonQuery();  } |