**Лабораторная работа №7**

**Работа с базами данных. OLE DB.**

*ODBC*

Интерфейс Microsoft Open Database Connectivity (ODBC) – интерфейс для языка программирования C, позволяющий приложениям получить доступ с различными СУБД. ODBC представляет собой низкоуровневый интерфейс, разработанный для реляционных БД и отличающийся высокой производительностью.

*OLE DB*

OLE DB (англ. Object Linking and Embedding, Database) — набор COM-интерфейсов, которые позволяют приложениям унифицировано работать с данными разных источников и хранилищ информации. Разработана Microsoft в качестве дальнейшего развития технологии доступа к данным и должен был прийти на замену и в качестве преемника ODBC, расширяя набор функций для поддержки более широкого круга нереляционных источников данных, таких как объектно-ориентированные базы данных или электронные таблицы, и для которых не обязательно использовать SQL.

OLE DB отделяет хранилище данных от приложения, которое должно иметь доступ к нему через набор абстракций, состоящий из источника данных (DataSource), сессии (Session), команды (Command) и набора строк (Rowset). Это было сделано для предоставления унифицированного доступа к различным видам и источникам данных и изоляцию специфики взаимодействия с конкретным хранилищем. OLE DB концептуально разделена на потребителей (клиентов) и поставщиков (провайдеров). Потребителем является приложение, которому необходим доступ к данным, а поставщик реализует интерфейс доступа к данным и, следовательно, обеспечивает информацией потребителя*.*

Пространство имен System.Data.OleDb является поставщиком данных .NET Framework для OLE DB.

|  |  |
| --- | --- |
| [OleDbCommand](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbcommand?view=netframework-4.8) | Представляет оператор SQL или хранимую процедуру, применяемую к источнику данных. |
| [OleDbCommandBuilder](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbcommandbuilder?view=netframework-4.8) | Автоматически генерирует однотабличные команды, которые позволяют согласовать изменения, вносимые в объект [DataSet](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.dataset?view=netframework-4.8), со связанной базой данных. Этот класс не наследуется. |
| [OleDbConnection](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbconnection?view=netframework-4.8) | Представляет открытое подключение к источнику данных. |
| [OleDbConnectionStringBuilder](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbconnectionstringbuilder?view=netframework-4.8) | Удобное средство, с помощью которого можно создавать используемые классом [OleDbConnection](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbconnection?view=netframework-4.8) строки соединений и управлять их содержимым. |
| [OleDbDataAdapter](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb.oledbdataadapter?view=netframework-4.8) | Представляет набор команд данных и подключение базы данных, которые используются для заполнения [DataSet](https://docs.microsoft.com/ru-ru/dotnet/api/system.data.dataset?view=netframework-4.8) и обновления источника данных. |

|  |
| --- |
| #Данный пример демонстрирует возможности использования OLE DB в приложении Windows Forms  string connString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\MyDB\_old.mdb";  OleDbDataAdapter oledbAdapter = new OleDbDataAdapter();  string sql = null;  try  {  if (string.IsNullOrEmpty(flightIDBox.Text) || string.IsNullOrEmpty(airlineBox.Text) || string.IsNullOrEmpty(originBox.Text) || string.IsNullOrEmpty(destinationBox.Text))  {  MessageBox.Show("Please give valid values!");  return;  }  connection = new OleDbConnection(connString);  #Добавление данных – команда INSERT. Аналогично SELECT, UPDATE, DELETE  sql = "insert into Flight (FlightID, AirlineCompany, Origin, Destination, DateAndTime) values (@id, @ac, @ori, @dest, @dt)";  connection.Open();  OleDbCommand cmd = new OleDbCommand(sql, connection);  cmd.Parameters.AddWithValue("@id", flightIDBox.Text);  cmd.Parameters.AddWithValue("@ac", airlineBox.Text);  cmd.Parameters.AddWithValue("@ori", originBox.Text);  cmd.Parameters.AddWithValue("@dest", destinationBox.Text);  cmd.Parameters.AddWithValue("@dt", DateTime.Parse(dateTimePicker.Value.ToString()));  cmd.ExecuteNonQuery();  MessageBox.Show("Row(s) Inserted! ");  }  catch (System.Data.OleDb.OleDbException ex)  {  MessageBox.Show(ex.ToString());  }  this.Close();  } |

**Варианты заданий**

## Задание

По заданию ЛР6 реализовать набор связанных таблиц в Microsoft Access (.mdb), соответствующую выданному варианту. Одна таблица соответствует одному классу. Базу данных заполнять непосредственно в MS Access.

Подключиться к базе данных (используя OleDbConnection – в строке подключения передать параметры Provider и Data Source).

Реализовать консольное приложение, осуществляющее поиск по существующей БД и выводящее релевантные данные на печать: например, найдя информацию о фирме, программа должна выводить на печать все данные об этой фирме. Поиск осуществляется посредством SQL-запросов к БД. Поиск осуществляется по всем полям.

Подсказка: для выполнения лабораторной нужно создать экземпляры классов из пространства имен System.Data.OleDb: OleDbDataAdapter, OleDbConnection, OleDbCommand.

Приложение 1

Справочные материалы

System.Data.OleDb Namespace

<https://docs.microsoft.com/ru-ru/dotnet/api/system.data.oledb?view=netframework-4.8>

C# Tutorials – Queries

<https://csharp-tutorials1.blogspot.com/2016/02/insert-update-delete-in-access-database.html>