

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data.OleDb;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using System.Data;
8
9 namespace Nihulon2.Model.DbAccess
10 {
11     // The class that provides the connection to the DB
12     // end execution of different queries
13     public class DbAccess
14     {
15         #region Constructor + Members
16         protected OleDbConnection _conn = null;
17
18         public DbAccess(string connectionString)
19         {
20             _conn = new OleDbConnection(connectionString);
21         }
22         #endregion
23
24         #region Protected Methods
25         // Open a connection with the DB
26         protected void Connect()
27         {
28             if(_conn.State != ConnectionState.Open)
29             {
30                 _conn.Open();
31             }
32         }
33         // Close the connection
34         protected void Disconnect()
35         {
36             _conn.Close();
37         }
38         // Execute a query that doesn't returns any result
39         protected void ExecuteSimpleQuery(OleDbCommand command)
40         {
41             lock(_conn){
42                 Connect();
43                 command.Connection = _conn;
44                 try
45                 {
46                     command.ExecuteNonQuery();
47                 }
48                 finally
49                 {
50                     Disconnect();
51                 }
52             }
53         }
54         // Execute a query that returns an integer value
55         protected int ExecuteScalarIntquery(OleDbCommand command)
56         {
```

```
57         int ret = -1;
58         lock (_conn)
59         {
60             Connect();
61             command.Connection = _conn;
62             try
63             {
64                 ret = (int)command.ExecuteScalar();
65             }
66             finally
67             {
68                 Disconnect();
69             }
70         }
71         return ret;
72     }
73     // Execute a query that returns complex result
74     protected DataSet GetMultipleQuery(OleDbCommand command)
75     {
76         DataSet dataset = new DataSet();
77         lock (_conn)
78         {
79             Connect();
80             command.Connection = _conn;
81             try
82             {
83                 OleDbDataAdapter adapter = new OleDbDataAdapter();
84                 adapter.SelectCommand = command;
85                 adapter.Fill(dataset);
86             }
87             finally
88             {
89                 Disconnect();
90             }
91         }
92         return dataset;
93     }
94     #endregion
95 }
96 }
97
98
```