

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Text;
7 using System.Windows.Forms;
8 using System.Data.OleDb;
9 using System.Diagnostics;
10
11 namespace OleDb2DataGrid
12 {
13     public partial class Form1 : Form
14     {
15         OleDbObject carbonDbConnect = null;
16         OleDbObject projDbConnect = null;
17         DataTable dt_carbon = null;
18         DataTable dt_proj = null;
19
20         public Form1()
21         {
22             InitializeComponent();
23             filllTable();
24         }
25
26         public void filllTable()
27         {
28             carbonDbConnect = new OleDbObject("carbon.mdb");
29             dt_carbon = carbonDbConnect.getDataTable("Select * FROM WallMapping;");
30             dataGridView1.DataSource = dt_carbon;
31
32             projDbConnect = new OleDbObject("classExercisell122009.mdb");
33             dt_proj = projDbConnect.getDataTable("Select * FROM Walls;");
34             dataGridView2.DataSource = dt_proj;
35
36             // create CO2 dictionary
37             getCarbonDictionary();
38         }
39
40         public void getCarbonDictionary()
41         {
42             string value = "";
43             // iterate through each rows of table
44
45             Dictionary<string, double> carbon = new Dictionary<string, double>();
46             value += ("Number of Rows in Carbon database: [" + dt_carbon.Rows.Count + "]\n");
47             value += ("Number of Rows in Project database: [" + dt_proj.Rows.Count + "]\n");
48
49             foreach (DataRow r in dt_carbon.Rows)
50             {
51                 string typeId = r["TypeId"].ToString();
52                 double CO2 = unitConvert.getDouble(r["CO2"]);
```

```
53
54     value += ("TypeId : " + typeId + ", ");
55     value += ("CO2 : " + CO2);
56     value += "\n";
57
58     Debug.Print("Key :[{0}] , Value:[{1}]", typeId, CO2);
59     if (!carbon.ContainsKey(typeId))
60     {
61         carbon.Add(typeId, CO2);
62     }
63 }
64 //MessageBox.Show(value);
65 Debug.Print(value);
66
67 int count = 0;
68 Dictionary<string, double> found = new Dictionary<string, double>();
69
70 foreach (DataRow r in dt_proj.Rows)
71 {
72
73     string typeId = r["TypeId"].ToString();
74
75     if (carbon.ContainsKey(typeId))
76     {
77         count++;
78         if (!found.ContainsKey(typeId))
79         {
80             double d;
81             carbon.TryGetValue(typeId, out d);
82             found.Add(typeId, d);
83             Debug.Print("TypeId:[{0}], CO2:[{1}]", typeId, d);
84         }
85     }
86
87 }
88
89 Debug.Print("Number of types in found: " + found.Keys.Count + "\n");
90 }
91
92
93 }
94
95 /// <summary>
96 /// OleDbObject to open an OleDbConnection
97 /// </summary>
98 public class OleDbObject
99 {
100     string filePath;
101     OleDbConnection dbConnection;
102     string ConnectionHeader = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=";
103
104 }
```

```
105     public OleDbObject(string p)
106     {
107         filePath = p;
108
109         //initiate OleDb connection
110         createOleDbConnection(ConnectionHeader + filePath);
111     }
112
113     private void createOleDbConnection(string connectionString)
114     {
115         try
116         {
117             dbConnection = new OleDbConnection(connectionString);
118             dbConnection.Open();
119         }
120         catch (Exception ex)
121         {
122             MessageBox.Show(ex.Message);
123             return;
124         }
125     }
126
127     public DataTable getDataTable(string queryString)
128     {
129         // initiate OleDbCommand and DataTable
130         OleDbCommand sqlCommand = new OleDbCommand(queryString, dbConnection);
131         DataTable dataTable = new DataTable();
132         OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand);
133
134         // fill in the DataTable
135         dataAdapter.Fill(dataTable);
136
137         return dataTable;
138     }
139
140     private int exeSqlCommand(string queryString, OleDbConnection db)
141     {
142         OleDbCommand command = new OleDbCommand(queryString, db);
143
144         // ExecuteNonQuery() : Executes commands such as SQL INSERT,
145         // DELETE, UPDATE, and SET statements.
146         return command.ExecuteNonQuery();
147     }
148
149 }
150
151 public class unitConvert
152 {
153     public static double getDouble(object obj)
154     {
155         try
156         {
```

```
157         return (double)obj;
158     }
159     catch
160     {
161         return 0.0;
162     }
163 }
164 }
165
166 public static string getString(object obj)
167 {
168     try
169     {
170         return (string)obj;
171     }
172     catch
173     {
174         return null;
175     }
176 }
177 }
178 }
179 }
```