For: TJ Liggett

Assignment: Family Database

TFS Name: \Assignments\9FamilyDB

Student: Please answer the questions, then use the Insert, Screenshot option in Word to snip an appropriate sample of your executing program’s output. Copy the code from your .cs file(s) into the code section below. Your code should match the code submitted in TFS.

Be sure to review your graded assignment for instructor comments!

|  |
| --- |
| **Analysis** |
| *Describe the problem, including input and output, in your own words* |
| I had to display family members inside of an access database. |

|  |
| --- |
| **Design** |
| *Describe the major steps for problem solving* |
| For the most part, I just incorporated what we learned in class. Though I did decide to dock the ListBox, so that it resizes with the window. |

|  |
| --- |
| **Testing** |
| *Describe your test plan* |
| As we did in class, I got an error message if my program wasn’t working correctly. I actually did have to install the access database engine distributable for the program to work on my computer at home. |

|  |
| --- |
| **Screenshot(s)** |
| *Paste screen shot(s) here, within this table entry* |
|  |

|  |
| --- |
| **Code** |
| *Paste code here, within this table entry. Use the retain formatting option of the Paste* |
| 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;  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* This program pulls data \*  \* from the access database.\*  \* Similar to what we did \*  \* in class. \*  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  namespace \_9FamilyDB  {  public partial class Form1 : Form  {  //sets up connection  private OleDbConnection dbConnection;  private string sConnection;    //holds command  private OleDbCommand dbCmd;  private string sql;    //sets up reader  private OleDbDataReader familyReader;  private Family familyObj;  public Form1()  {  InitializeComponent();  }  private void Form1\_Load(object sender, EventArgs e)  {  try  {  sConnection =  "Provider=Microsoft.ACE.OLEDB.12.0;" +  "Data Source=familyDB.accdb;";  dbConnection = new OleDbConnection(sConnection);  dbConnection.Open();  //build our sql statement  sql = "SELECT \* FROM familyTable " +  "ORDER BY LastName ASC, FirstName ASC;";  //instance of cmd is created  dbCmd = new OleDbCommand();  dbCmd.CommandText = sql;  dbCmd.Connection = dbConnection;  //create reader object  familyReader = dbCmd.ExecuteReader();  while (familyReader.Read())  {  familyObj = new Family(familyReader["FirstName"].ToString(),  familyReader["LastName"].ToString(),  familyReader["Relationship"].ToString(),  familyReader["Hometown"].ToString()  );  lstFamily.Items.Add(familyObj);  }  familyReader.Close();  }  catch (System.Exception exc)  {  lblMessage.Text = exc.Message;  }  }  }  } |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_9FamilyDB//.Business\_Logic  {  class Family  {  //private class variables  private int famID;  private string firstName;  private string lastName;  private string relationship;  private string hometown;  public int age;  public Family()  {  }  public Family(string fName, string lName,  string relation, string town)  {  this.firstName = fName;  this.lastName = lName;  this.relationship = relation;  this.hometown = town;  }  public Family(int ID, string fName, string lName,  string relation, string town, int age)  {  this.famID = ID;  this.firstName = fName;  this.lastName = lName;  this.relationship = relation;  this.hometown = town;  this.age = age;  }  #region getters/setters  public int FamID  {  get  {  return famID;  }  }  public string FirstName  {  get  {  return firstName;  }  set  {  firstName = value;  }  }  public string LastName  {  get  {  return lastName;  }  set  {  lastName = value;  }  }  public string Relationship  {  get  {  return relationship;  }  set  {  relationship = value;  }  }  public string Hometown  {  get  {  return hometown;  }  set  {  hometown = value;  }  }  public int Age  {  get  {  return age;  }  set  {  age = value;  }  }  #endregion  public override string ToString()  {  return firstName + " " + lastName + ", "  + relationship + ", " + hometown;  }  }  } |