Ders İşlemleri'nin kodları:

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;

namespace SınavListesi

{

public partial class DersIslemleri : Form

{

public DersIslemleri()

{

InitializeComponent();

}

private void DersIslemleri\_Load(object sender, EventArgs e)

{

VeriTabaniIslemleri v = new VeriTabaniIslemleri(Properties.Settings.Default.VeriConnectionString);

string Dersler = v.DersListesi2();

string[] DerslerD = Dersler.Split();

for (int i = 0; i < DerslerD.Length; i++)

{

listBox1.Items.Add(DerslerD[i]);

}

}

}

}

Form1'in kodları:

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;

namespace SınavListesi

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void ÖğrenciİşlemleriToolStripMenuItem\_Click(object sender, EventArgs e)

{

Öğrenci ogr = new Öğrenci();

if (Application.OpenForms["Öğrenci"] != null)

{

ogr.Activate();

}

else

{

ogr.MdiParent = this;

ogr.Show();

}

}

private void SınıfListesiOluşturToolStripMenuItem\_Click(object sender, EventArgs e)

{

SınıfListesi ogr = new SınıfListesi();

if (Application.OpenForms["SınıfListesi"] != null)

{

ogr.Activate();

}

else

{

ogr.MdiParent = this;

ogr.Show();

}

}

private void SınıfİşlemleriToolStripMenuItem\_Click(object sender, EventArgs e)

{

SinifIslemleri goster = new SinifIslemleri();

goster.MdiParent = this;

goster.Show();

}

private void ÇıkışToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void HakkındaToolStripMenuItem\_Click(object sender, EventArgs e)

{

Hakkında goster = new Hakkında();

goster.MdiParent = this;

goster.Show();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void DersİşlemleriToolStripMenuItem\_Click(object sender, EventArgs e)

{

DersIslemleri goster = new DersIslemleri();

goster.MdiParent = this;

goster.Show();

}

private void YardımToolStripMenuItem\_Click(object sender, EventArgs e)

{

/\*OpenFileDialog dosya = new OpenFileDialog();

dosya.FileName = "Yardım-Kullanma Klavuzu";

dosya.Filter= "Pdf Files|\*.pdf";

dosya.ShowDialog();

Bu komutlarla açması gerekirken açmadı.Bu kısımda nerede eksik yaptığımı bulamadım.\*/

}

}

}

VeriTabaniIslemleri'nin kodları:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Data;

using System.Windows.Forms;

using System.Collections;

namespace SınavListesi

{

class VeriTabaniIslemleri

{

OleDbConnection baglanti;

public VeriTabaniIslemleri(string ConnectionString)

{

baglanti = new OleDbConnection(ConnectionString);

}

public void OgrenciSil()

{

OleDbCommand komut = new OleDbCommand("Delete from Ogrenciler", baglanti);

baglanti.Open();

komut.ExecuteNonQuery();

baglanti.Close();

}

public bool OgrenciVarMi(string OgrNo, string DersNo)

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select Ad from Ogrenciler where ((Ogrenciler.OgrNo=?)and(Ogrenciler.DersNo=?));", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

komut.Parameters.AddWithValue("OgrNo", OgrNo);

komut.Parameters.AddWithValue("DersNo", DersNo);

OleDbDataReader oku = komut.ExecuteReader();

if (oku.HasRows)

{

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return true;

}

else

{

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return false;

}

}

public bool DersVarMi(string DersNo)

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select DersNo from Dersler where (Dersler.DersNo=?);", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

komut.Parameters.AddWithValue("DersNo", DersNo);

OleDbDataReader oku = komut.ExecuteReader();

if (oku.HasRows)

{

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return true;

}

else

{

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return false;

}

}

public void DersEkle(string DersNo, string DersAdi = "Dersin Adi Verilmemiş")

{

if (DersVarMi(DersNo) == false)

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Insert into Dersler(DersNo,DersAdi) Values(?,?);", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

komut.Parameters.AddWithValue("DersNo", DersNo);

komut.Parameters.AddWithValue("DersAdi", DersAdi);

komut.ExecuteNonQuery();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

}

}

public string OgrYukle(string[] veri, ToolStripProgressBar p)

{

string HataliSatirlar = "Dosyadan veritabanına aktarılamayan satırlar\n";

baglanti.Open();

p.Maximum = veri.Length;

for (int i = 1; i < veri.Length; i++)

{

p.Value = i;

OleDbCommand komut = new OleDbCommand("Insert into Ogrenciler(OgrNo,Ad,Soyad,DersNo,SinifNo) Values(?,?,?,?,?);", baglanti);

string[] veri2 = veri[i].Replace('\r', ' ').Split(';');

if (veri2.Length >= 4)

{

if (OgrenciVarMi(veri2[0], veri2[3]) == false)

{

komut.Parameters.AddWithValue("OgrNo", veri2[0]);

komut.Parameters.AddWithValue("Ad", veri2[1]);

komut.Parameters.AddWithValue("Soyad", veri2[2]);

komut.Parameters.AddWithValue("DersNo", veri2[3]);

komut.Parameters.AddWithValue("SinifNo", "1");

DersEkle(veri2[3]);

komut.ExecuteNonQuery();

komut.Dispose();

}

else

{

HataliSatirlar = HataliSatirlar + i + "\t" + "Tekrarlı Veri " + veri[i] + "\n";

}

}

else

{

HataliSatirlar = HataliSatirlar + i + "\t" + "Eksik Veri " + veri[i] + "\n";

}

}

baglanti.Close();

return HataliSatirlar;

}

public int DersiAlanOgrenciSayisi(string DersNo)

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select Count (\*) from Ogrenciler Where (Ogrenciler.DersNo=?);", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

komut.Parameters.AddWithValue("DersNo", DersNo);

int toplam = Convert.ToInt16(komut.ExecuteScalar());

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return toplam;

}

public string DersListesi()

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select \* from Dersler ;", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

OleDbDataReader oku = komut.ExecuteReader();

string Cevap = "";

while (oku.Read())

{

Cevap = Cevap + oku["DersNo"] + ":" + oku["DersAdi"] + ":" + DersiAlanOgrenciSayisi(oku["DersNo"].ToString()) + "\n";

}

Cevap = Cevap.Remove(Cevap.Length - 1);

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return Cevap;

}

public string SinifListesi()

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select \* from Siniflar ;", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

OleDbDataReader oku = komut.ExecuteReader();

string Cevap = "";

while (oku.Read())

{

Cevap = Cevap + oku["SinifNo"] + ":" + oku["SinifAdi"] + ":" +oku["Kontejan"]+ "\n";

}

Cevap = Cevap.Remove(Cevap.Length - 1);

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return Cevap;

}

public void SinifDoldur(string DersNo,ArrayList Siniflar)

{

ArrayList Ogrenciler = new ArrayList();

OleDbCommand komut = new OleDbCommand("Select OgrNo from Ogrenciler Where (Ogrenciler.DersNo=?);", baglanti);

baglanti.Open();

komut.Parameters.AddWithValue("DersNo", DersNo);

OleDbDataReader oku = komut.ExecuteReader();

while (oku.Read())

{

Ogrenciler.Add(oku["OgrNo"]);

}

oku.Close();

Random r = new Random();

for(int i = 0; i < Siniflar.Count; i++)

{

string[] Sinif = (string[])Siniflar[i];

int kontejan = Convert.ToInt16(Sinif[2]);

for(int j = 0; j < kontejan; j++)

{

if (Ogrenciler.Count == 0)

break;

int satirno = r.Next(Ogrenciler.Count);

string OgrNo = Ogrenciler[satirno].ToString();

Ogrenciler.RemoveAt(satirno);

OleDbCommand komut2 = new OleDbCommand("UPDATE Ogrenciler Set SinifNo=? Where ((Ogrenciler.OgrNo=?) and (Ogrenciler.DersNo=?));", baglanti);

komut2.Parameters.AddWithValue("SinifNo", Sinif[0]);

komut2.Parameters.AddWithValue("OgrNo", OgrNo);

komut2.Parameters.AddWithValue("DersNo", DersNo);

komut2.ExecuteNonQuery();

komut2.Dispose();

}

}

baglanti.Close();

}

public string SinifListesi2()

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select \* from Siniflar ;", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

OleDbDataReader oku = komut.ExecuteReader();

string Cevap = "";

while (oku.Read())

{

Cevap = Cevap + oku["SinifAdi"]+" ";

}

Cevap = Cevap.Remove(Cevap.Length - 1);

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return Cevap;

}

public string DersListesi2()

{

bool BuMetodKullandi = false;

OleDbCommand komut = new OleDbCommand("Select \* from Dersler ;", baglanti);

if (baglanti.State == ConnectionState.Closed)

{

baglanti.Open();

BuMetodKullandi = true;

}

OleDbDataReader oku = komut.ExecuteReader();

string Cevap = "";

while (oku.Read())

{

Cevap = Cevap + oku["DersNo"] + oku["DersAdi"] + " ";

}

Cevap = Cevap.Remove(Cevap.Length - 1);

oku.Close();

komut.Dispose();

if (BuMetodKullandi)

baglanti.Close();

return Cevap;

}

}

}

SinifIslemleri'nin kodları:

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;

namespace SınavListesi

{

public partial class SinifIslemleri : Form

{

public SinifIslemleri()

{

InitializeComponent();

}

private void SinifIslemleri\_Load(object sender, EventArgs e)

{

VeriTabaniIslemleri v = new VeriTabaniIslemleri(Properties.Settings.Default.VeriConnectionString);

string Siniflar = v.SinifListesi2();

string[] SiniflarD = Siniflar.Split();

for (int i = 0; i < SiniflarD.Length; i++)

{

listBox1.Items.Add(SiniflarD[i]);

}

}

}

}

SınıfListesi'nin kodları:

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.Collections;

namespace SınavListesi

{

public partial class SınıfListesi : Form

{

public SınıfListesi()

{

InitializeComponent();

}

int DerstekiOgrenciSayisi = 0;

int SeciliSinifKontejan = 0;

private void SınıfListesi\_Load(object sender, EventArgs e)

{

VeriTabaniIslemleri v = new VeriTabaniIslemleri(Properties.Settings.Default.VeriConnectionString);

string[] Dersler = v.DersListesi().Split('\n');

listBox1.Items.AddRange(Dersler);

string[] Siniflar = v.SinifListesi().Split('\n');

checkedListBox1.Items.AddRange(Siniflar);

}

private void ListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

string[] a = listBox1.Items[listBox1.SelectedIndex].ToString().Split(':');

DerstekiOgrenciSayisi = Convert.ToInt16(a[2]);

label2.Text = a[2];

}

private void CheckedListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

SeciliSinifKontejan = 0;

for(int i = 0; i < checkedListBox1.CheckedItems.Count; i++)

{

string[] a = checkedListBox1.CheckedItems[i].ToString().Split(':');

SeciliSinifKontejan = SeciliSinifKontejan + Convert.ToInt16(a[2]);

}

label4.Text = SeciliSinifKontejan.ToString();

}

private void Button1\_Click(object sender, EventArgs e)

{

VeriTabaniIslemleri v = new VeriTabaniIslemleri(Properties.Settings.Default.VeriConnectionString);

try

{

string[] a = listBox1.Items[listBox1.SelectedIndex].ToString().Split(':');

string DersNo = a[0];

ArrayList Siniflar = new ArrayList();

for (int i = 0; i < checkedListBox1.CheckedItems.Count; i++)

{

string[] SinifListesi = checkedListBox1.CheckedItems[i].ToString().Split(':');

Siniflar.Add(SinifListesi);

}

v.SinifDoldur(DersNo, Siniflar);

}

catch

{

MessageBox.Show("Lütfen ders seçiniz!");

}

try

{

int L = Convert.ToInt16(label2.Text);

int C = Convert.ToInt16(label4.Text);

if (L <= C)

{

MessageBox.Show(listBox1.Items[listBox1.SelectedIndex] + " " + "Dersinin Sınıf Ataması Başarı ile Gerçekleşmiştir");

}

else if (L > C)

{

int fark = L - C;

MessageBox.Show(listBox1.Items[listBox1.SelectedIndex] + " " + "Dersinin Sınıf Ataması Başarı ile Gerçekleşmiştir" + " " + fark + " " + "kişi ataması yapılamamıştır");

}

}

catch

{

}

}

}

}

Öğrenci'nin kodları:

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.IO;

namespace SınavListesi

{

public partial class Öğrenci : Form

{

public Öğrenci()

{

InitializeComponent();

}

private void OgrencilerBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.ogrencilerBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.veriDataSet);

}

private void Öğrenci\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'veriDataSet.Dersler' table. You can move, or remove it, as needed.

this.derslerTableAdapter.Fill(this.veriDataSet.Dersler);

// TODO: This line of code loads data into the 'veriDataSet.Siniflar' table. You can move, or remove it, as needed.

this.siniflarTableAdapter.Fill(this.veriDataSet.Siniflar);

// TODO: This line of code loads data into the 'veriDataSet.Ogrenciler' table. You can move, or remove it, as needed.

this.ogrencilerTableAdapter.Fill(this.veriDataSet.Ogrenciler);

}

private void ToolStripButton1\_Click(object sender, EventArgs e)

{

VeriTabaniIslemleri v = new VeriTabaniIslemleri(Properties.Settings.Default.VeriConnectionString);

if (DialogResult.OK == MessageBox.Show("Bu işlem öğrenciler tablosunu silecektir.Onaylıyor musunuz?", "Tablo Silme", MessageBoxButtons.OKCancel, MessageBoxIcon.Question))

{

v.OgrenciSil();

Öğrenci\_Load(sender, e);

}

}

private void ToolStripButton2\_Click(object sender, EventArgs e)

{

VeriTabaniIslemleri v = new VeriTabaniIslemleri(Properties.Settings.Default.VeriConnectionString);

OpenFileDialog op = new OpenFileDialog();

if (op.ShowDialog() == DialogResult.OK)

{

StreamReader oku = new StreamReader(op.FileName, Encoding.GetEncoding("iso-8859-9"));

string[] veri = oku.ReadToEnd().Split('\n');

string cevap= v.OgrYukle(veri, toolStripProgressBar1);

Öğrenci\_Load(sender, e);

MessageBox.Show(cevap);

toolStripProgressBar1.Value = 0;

}

}

}

}