Video

**DataUtil**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml;

using System.IO;

using System.Windows.Forms;

namespace Video3

{

class DataUtil

{

XmlDocument doc;

XmlElement root;

string filename;

public DataUtil()

{

doc = new XmlDocument();

filename = "Course.xml";

if (!File.Exists(filename))

{

XmlElement course = doc.CreateElement("lophoc");

doc.AppendChild(course);

doc.Save(filename);

}

doc.Load(filename);

root = doc.DocumentElement;

}

public void AddStudent(Student s)

{

XmlNode sid = root.SelectSingleNode("student[@id='" + s.id + "']");

if (sid != null)

{

MessageBox.Show("Trùn mã id " + s.id);

return;

}

else

{

XmlElement st = doc.CreateElement("student");

st.SetAttribute("id", s.id);

XmlElement name = doc.CreateElement("name");

name.InnerText = s.name;

XmlElement age = doc.CreateElement("age");

age.InnerText = s.age;

XmlElement city = doc.CreateElement("city");

city.InnerText = s.city;

st.AppendChild(name);

st.AppendChild(age);

st.AppendChild(city);

root.AppendChild(st);

doc.Save(filename);

}

}

public bool UpdateStudent(Student s)

{

XmlNode stfind = root.SelectSingleNode("student[@id='" + s.id + "']");

if (stfind != null)

{

XmlElement st = doc.CreateElement("student");

st.SetAttribute("id", s.id);

XmlElement name = doc.CreateElement("name");

name.InnerText = s.name;

XmlElement age = doc.CreateElement("age");

age.InnerText = s.age;

XmlElement city = doc.CreateElement("city");

city.InnerText = s.city;

st.AppendChild(name);

st.AppendChild(age);

st.AppendChild(city);

root.ReplaceChild(st, stfind);

doc.Save(filename);

return true;

}

return false;

}

public List<Student> GetAllStudent()

{

XmlNodeList nodes = root.SelectNodes("student");

List<Student> li = new List<Student>();

foreach(XmlNode item in nodes)

{

Student s = new Student();

s.id = item.Attributes[0].InnerText;

s.name = item.SelectSingleNode("name").InnerText;

s.age = item.SelectSingleNode("age").InnerText;

s.city = item.SelectSingleNode("city").InnerText;

li.Add(s);

}

return li;

}

public bool DeleteStudent(string id)

{

XmlNode stfind = root.SelectSingleNode("student[@id='" + id + "']");

if (stfind != null)

{

root.RemoveChild(stfind);

doc.Save(filename);

return true;

}

return false;

}

public Student FindById(string id)

{

XmlNode stfind = root.SelectSingleNode("student[@id='" + id + "']");

Student s = null;

if (stfind != null)

{

s = new Student();

s.id = stfind.Attributes[0].InnerText;

s.name = stfind.SelectSingleNode("name").InnerText;

s.age = stfind.SelectSingleNode("age").InnerText;

s.city = stfind.SelectSingleNode("city").InnerText;

}

return s;

}

public List<Student> FindByCity(string city)

{

XmlNodeList students = root.SelectNodes("student[city='" + city + "']");

List<Student> li = new List<Student>();

foreach (XmlNode item in students)

{

Student s = new Student();

s.id = item.Attributes[0].InnerText;

s.name = item.SelectSingleNode("name").InnerText;

s.age = item.SelectSingleNode("age").InnerText;

s.city = item.SelectSingleNode("city").InnerText;

li.Add(s);

}

return li;

}

}

}

**Form1.cs**

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 Video3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

DataUtil data = new DataUtil();

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

{

DisplayData();

}

private void DisplayData()

{

dataGridView1.DataSource = data.GetAllStudent();

dataGridView1.Columns[0].Width = 50;

dataGridView1.Columns[1].Width = 250;

dataGridView1.Columns[2].Width = 50;

dataGridView1.Columns[3].Width = 200;

dataGridView1.Columns[0].Name = "ID";

dataGridView1.Columns[1].Name = "Name";

dataGridView1.Columns[2].Name = "Age";

dataGridView1.Columns[3].Name = "City";

lblCount.Text = dataGridView1.Rows.Count + "";

}

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

{

Student s = new Student();

s.id = txtId.Text;

s.name = txtName.Text;

s.age = txtAge.Text;

s.city = txtCity.Text;

data.AddStudent(s);

ClearTextboxs();

DisplayData();

}

private void ClearTextboxs()

{

txtId.Clear();

txtName.Clear();

txtAge.Clear();

txtCity.Clear();

ActiveControl = txtId;

}

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

{

Close();

}

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

{

DisplayData();

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

Student s = (Student)dataGridView1.CurrentRow.DataBoundItem;

txtId.Text = s.id;

txtName.Text = s.name;

txtAge.Text = s.age;

txtCity.Text = s.city;

}

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

{

Student s = new Student();

s.id = txtId.Text;

s.name = txtName.Text;

s.age = txtAge.Text;

s.city = txtCity.Text;

bool kt = data.UpdateStudent(s);

if (!kt)

{

MessageBox.Show("Không cập nhật được sinh viên có id " + s.id);

return;

}

DisplayData();

}

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

{

ClearTextboxs();

}

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

{

DialogResult d = MessageBox.Show("Bạn có chắc chắn xóa không?","Thông báo",MessageBoxButtons.YesNo,MessageBoxIcon.Error);

if (d == DialogResult.Yes)

{

bool kt = data.DeleteStudent(txtId.Text);

if (!kt)

{

MessageBox.Show("Có lỗi khi xóa", "Thông báo");

return;

}

DisplayData();

ClearTextboxs();

}

}

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

{

string id = txtId.Text;

List<Student> li = new List<Student>();

Student s = data.FindById(id);

if (s != null)

{

li.Add(s);

dataGridView1.DataSource = li;

lblCount.Text = dataGridView1.Rows.Count + "";

txtId.Text = s.id;

txtName.Text = s.name;

txtAge.Text = s.age;

txtCity.Text = s.city;

}

else

{

MessageBox.Show("Không có sinh viên có mã = " + id, "Thông báo");

}

}

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

{

string city = txtCity.Text;

dataGridView1.DataSource = data.FindByCity(city);

lblCount.Text = dataGridView1.Rows.Count + "";

ClearTextboxs();

txtCity.Text = city;

}

}

}

Datautil

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Windows.Forms;

using System.Xml;

namespace Deso3

{

class DataUtil

{

XmlDocument doc;

XmlElement root;

string filename;

public DataUtil()

{

doc = new XmlDocument();

filename="khachhang.xml";

if (!File.Exists(filename))

{

XmlElement khachhang = doc.CreateElement("danhsachkhachhang");

doc.AppendChild(khachhang);

doc.Save(filename);

}

doc.Load(filename);

root = doc.DocumentElement;

}

public void AddCustomer(Customer c)

{

XmlNode cfind = root.SelectSingleNode("khachhang[@makh='" + c.makh + "']");

if (cfind != null)

{

MessageBox.Show("Trùng mã khách hàng " + c.makh);

return;

}

else

{

XmlElement kh = doc.CreateElement("khachhang");

kh.SetAttribute("chinhanh", c.chinhanh);

kh.SetAttribute("makh", c.makh);

XmlElement hoten = doc.CreateElement("hoten");

hoten.InnerText = c.hoten;

XmlElement diachi = doc.CreateElement("diachi");

diachi.InnerText = c.diachi;

XmlElement sodt = doc.CreateElement("sodt");

sodt.InnerText = c.sodt;

kh.AppendChild(hoten);

kh.AppendChild(diachi);

kh.AppendChild(sodt);

root.AppendChild(kh);

doc.Save(filename);

}

}

public List<Customer> GetAllCustomer()

{

XmlNodeList nodes = root.SelectNodes("khachhang");

List<Customer> li = new List<Customer>();

foreach(XmlNode item in nodes)

{

Customer c = new Customer();

c.chinhanh = item.Attributes[0].InnerText;

c.makh = item.Attributes[1].InnerText;

c.hoten = item.SelectSingleNode("hoten").InnerText;

c.diachi = item.SelectSingleNode("diachi").InnerText;

c.sodt = item.SelectSingleNode("sodt").InnerText;

li.Add(c);

}

return li;

}

public bool EditCustomer(Customer c)

{

XmlNode cfind = root.SelectSingleNode("khachhang[@makh='" + c.makh + "']");

if (cfind != null)

{

XmlElement kh = doc.CreateElement("khachhang");

kh.SetAttribute("chinhanh", c.chinhanh);

kh.SetAttribute("makh", c.makh);

XmlElement hoten = doc.CreateElement("hoten");

hoten.InnerText = c.hoten;

XmlElement diachi = doc.CreateElement("diachi");

diachi.InnerText = c.diachi;

XmlElement sodt = doc.CreateElement("sodt");

sodt.InnerText = c.sodt;

kh.AppendChild(hoten);

kh.AppendChild(diachi);

kh.AppendChild(sodt);

root.ReplaceChild(kh, cfind);

doc.Save(filename);

return true;

}

return false;

}

public bool DeleteCustomer(string makh)

{

XmlNode cfind = root.SelectSingleNode("khachhang[@makh='" + makh + "']");

if (cfind != null)

{

root.RemoveChild(cfind);

doc.Save(filename);

return true;

}

return false;

}

}

}

Form

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.Text.RegularExpressions;

namespace Deso3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

DataUtil data = new DataUtil();

private void dtgrid\_CellClick(object sender, DataGridViewCellEventArgs e)

{

Customer c = (Customer)dtgrid.CurrentRow.DataBoundItem;

cbBox1.Text = c.chinhanh;

txtmakh.Text = c.makh;

txthoten.Text = c.hoten;

txtdiachi.Text = c.diachi;

txtsodt.Text = c.sodt;

}

private void Display()

{

dtgrid.DataSource = data.GetAllCustomer();

dtgrid.Columns[0].HeaderText = "Chi nhánh";

dtgrid.Columns[1].HeaderText = "Mã KH";

dtgrid.Columns[2].HeaderText = "Họ tên";

dtgrid.Columns[3].HeaderText = "Địa chỉ";

dtgrid.Columns[4].HeaderText = "Số điện thoại";

}

private void ClearTextBox()

{

cbBox1.SelectedIndex = -1;

txtmakh.Clear();

txthoten.Clear();

txtdiachi.Clear();

txtsodt.Clear();

ActiveControl = txtmakh;

}

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

{

string makh = txtmakh.Text;

string hoten = txthoten.Text;

string diachi = txtdiachi.Text;

string sodt = txtsodt.Text;

string err = "";

if (cbBox1.SelectedIndex == -1)

{

err = "\n Bạn phải chọn chi nhánh";

}

if (String.IsNullOrEmpty(makh))

{

err += "\n Bạn phải nhập mã kh";

}

if (String.IsNullOrEmpty(hoten))

{

err += "\n Bạn phải nhập họ tên";

}

if (String.IsNullOrEmpty(diachi))

{

err += "\n Bạn phải nhập địa chỉ";

}if (!Regex.IsMatch(sodt,@"\d+"))

{

err += "\n Bạn phải nhập số điện thoại và là số";

}

if (err != "")

{

MessageBox.Show("Có lỗi xảy ra: " + err, "Thông báo lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

else

{

Customer c = new Customer();

c.chinhanh = cbBox1.SelectedItem.ToString();

c.makh = makh;

c.hoten = hoten;

c.diachi = diachi;

c.sodt = sodt;

data.AddCustomer(c);

ClearTextBox();

Display();

}

}

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

{

Customer c = new Customer();

c.chinhanh = cbBox1.SelectedItem.ToString();

c.makh = txtmakh.Text;

c.hoten = txthoten.Text;

c.diachi = txtdiachi.Text;

c.sodt = txtsodt.Text ;

bool k = data.EditCustomer(c);

if (!k)

{

MessageBox.Show("Không tồn tại mã khách hàng ","Thông báo lỗi");

return;

}

ClearTextBox();

Display();

}

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

{

DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn xóa?", "Thông báo xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

bool k = data.DeleteCustomer(txtmakh.Text);

if (!k)

{

MessageBox.Show("Không tồn tại mã khách hàng ", "Thông báo lỗi");

return;

}

ClearTextBox();

Display();

}

}

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

{

Close();

}

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

{

Display();

}

}

}