

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.Xml;

namespace Lab2

{

public partial class Form1 : Form

{

/\* interface IStrategy

{

List<Sportsman> Algorithm(Sportsman p, string path);

}

public static List<Sportsman> AllSportsmans(XmlDocument doc)

{

List<Sportsman> data2 = new List<Sportsman>();

XmlNodeList elem = doc.SelectNodes("//student");

try

{

foreach (XmlNode el in elem)

{

data2.Add(Info(el));

}

}

catch { }

return data2;

}

public static List<Sportsman> SearchByParam(string nodename, string val, string param, XmlDocument doc, int n)

{

List<Sportsman> sportsman = new List<Sportsman>();

if (param != String.Empty && param != null)

{

switch (n)

{

case 0:

{

XmlNodeList elem = doc.SelectNodes("//" + nodename + "[@" + val + "=\"" + param + "\"]");

try

{

foreach (XmlNode e in elem)

{

XmlNodeList list1 = e.ChildNodes;

foreach (XmlNode el in list1)

{

XmlNodeList list2 = el.ChildNodes;

foreach(XmlNode el1 in list2)

{

sportsman.Add(Info(el1));

}

}

}

}

catch { }

return sportsman;

}

case 1:

{

XmlNodeList elem = doc.SelectNodes("//" + nodename + "[@" + val + "=\"" + param + "\"]");

try

{

foreach (XmlNode e in elem)

{

XmlNodeList list1 = e.ChildNodes;

foreach (XmlNode el in list1)

{

sportsman.Add(Info(el));

}

}

}

catch { }

return sportsman;

}

case 2:

{

XmlNodeList elem = doc.SelectNodes("//" + nodename + "[@" + val + "=\"" + param + "\"]");

try

{

foreach (XmlNode e in elem)

{

sportsman.Add(Info(e));

}

}

catch { }

return sportsman;

}

default: break;

}

}

}

public static List<Sportsman> Cross( List<List<Sportsman> > list)

{

List<Sportsman> result = new List<Sportsman>();

try

{

if (list != null)

{

Sportsman[] st = list[0].ToArray();

if (st != null)

{

foreach(Sportsman elem in st)

{

bool IsIn = true;

foreach(List<Sportsman> t in list)

{

if (t.Count <= 0) return new List<Sportsman>();

foreach(Sportsman s in t)

{

IsIn = false;

if (elem.Comparing(s))

{

IsIn = true;

break;

}

}

if (!IsIn) break;

}

if (IsIn)

{

result.Add(elem);

}

}

}

}

}

catch { }

return result;

}

class Sportsman

{

public string Section = null;

public string Name = null;

public string Surname = null;

public string Faculty = null;

public string Visitor = null;

public string Schedule = null;

public string Competition = null;

public Sportsman() { }

public Sportsman(string[] data)

{

Section = data[0];

Name = data[1];

Surname = data[2];

Faculty = data[3];

Visitor = data[4];

Schedule = data[5];

Competition = data[6];

}

public Sportsman(IStrategy algo)

{

Section = String.Empty;

Name = String.Empty;

Surname = String.Empty;

Faculty = String.Empty;

Visitor = String.Empty;

Schedule = String.Empty;

Competition = String.Empty;

}

public bool Comparing(Sportsman sportsman)

{

if ((this.Section == sportsman.Section) &&

(this.Name == sportsman.Name) &&

(this.Surname == sportsman.Surname) &&

(this.Faculty == sportsman.Faculty) &&

(this.Visitor == sportsman.Visitor) &&

(this.Schedule == sportsman.Schedule) &&

(this.Competition == sportsman.Competition))

return true;

else return false;

}

public IStrategy Algo { get; set; }

public List<Sportsman> Algorithm(Sportsman param, string path)

{

return Algo.Algorithm(param, path);

}

}

private Sportsman OurSportsman()

{

string[] info = new string[6];

if (checkBox1.Checked) info[0] = Convert.ToString(checkBox1.Text);

if (checkBox2.Checked) info[1] = Convert.ToString(checkBox2.Text);

if (checkBox3.Checked) info[2] = Convert.ToString(checkBox3.Text);

if (checkBox4.Checked) info[3] = Convert.ToString(checkBox4.Text);

if (checkBox5.Checked) info[4] = Convert.ToString(checkBox5.Text);

if (checkBox6.Checked) info[5] = Convert.ToString(checkBox6.Text);

if (checkBox7.Checked) info[6] = Convert.ToString(checkBox7.Text);

Sportsman IdealSportsman = new Sportsman(info);

return IdealSportsman;

}

public List<Sportsman> Algorithm(Sportsman sportsman, string path)

{

info.Clear();

List<Sportsman> result = new List<Sportsman>();

Sportsman st = null;

string \_section = null;

string \_visitor = null;

while (BestReader.Read())

{

if(BestReader.Name == "section")

{

while (BestReader.MoveToNextAttribute())

{

if(BestReader.Name == "SECTION")

{

\_section = BestReader.Value;

}

}

}

if (BestReader.Name == "visitor")

{

while (BestReader.MoveToNextAttribute())

{

if (BestReader.Name == "VISITOR")

{

\_visitor = BestReader.Value;

}

}

}

if (BestReader.Name == "")

{

if(st == null)

{

st = new Sportsman();

st.Section = \_section;

st.Visitor = \_visitor;

}

else

{

st = new Sportsman();

st.Section = \_section;

st.Visitor = \_visitor;

}

if (BestReader.HasAttributes)

{

while (BestReader.MoveToNextAttribute())

{

if(BestReader.Name == "NAME")

{

st.Name = BestReader.Value;

}

if (BestReader.Name == "SURNAME")

{

st.Surname = BestReader.Value;

}

if (BestReader.Name == "FACULTY")

{

st.Faculty = BestReader.Value;

}

if (BestReader.Name == "SCHEDULE")

{

st.Schedule = BestReader.Value;

}

if (BestReader.Name == "COMPETITION")

{

st.Competition = BestReader.Value;

}

}

}

if(st.Surname != null)

{

result.Add(st);

}

}

}

info = Filtr(result, sportsman);

return Info();

}

private List<Sportsman> Info()

{

throw new NotImplementedException();

}

public List<Sportsman> Filtr(List<Sportsman> allStud, Sportsman param)

{

List<Sportsman> result = new List<Sportsman>();

if (allStud != 0)

{

foreach (Sportsman e in allStud)

{

try

{

if(

(e.Section == param.Section || param.Section == null) &&

(e.Name == param.Name || param.Name == null) &&

(e.Surname == param.Surname || param.Surname == null) &&

(e.Faculty == param.Faculty || param.Faculty == null) &&

(e.Schedule == param.Schedule || param.Schedule == null) &&

(e.Competition == param.Competition || param.Competition == null)

)

{

result.Add(e);

}

}

catch { }

}

}

return result;

}

string path = "DataBase.xml";

List<Sportsman> result = new List<Sportsman>();

private void Output(List<Sportsman> result)

{

int i = 1

Console.WriteLine("Alg +");

foreach (Sportsman n in result)

{

richTextBox1.AppendText(i++ + "." + "\n");

richTextBox1.AppendText("Section: "+ n.Section + "\n");

richTextBox1.AppendText("Name: " + n.Name + "\n");

richTextBox1.AppendText("Surname: " + n.Surname + "\n");

richTextBox1.AppendText("Faculty: " + n.Faculty + "\n");

richTextBox1.AppendText("Visitor"+ n.Visitor + "\n");

richTextBox1.AppendText("Schedule :"+ n.Schedule + "\n");

richTextBox1.AppendText("Competition"+ n.Competition + "\n");

richTextBox1.AppendText("-------------------------------------------------\n");

}

}

public void GetAllSportsmans()

{

XmlDocument doc = new XmlDocument();

doc.Load("DataBase.xml");

XmlNodeList elem = doc.SelectNodes("//section");

foreach (XmlNode e in elem)

{

XmlNodeList list1 = e.ChildNodes;

foreach (XmlNode el in list1)

{

XmlNodeList list2 = el.ChildNodes;

foreach (XmlNode el1 in list2)

{

string section = el1.ParentNode.ParentNode.Attributes.GetNamedItem("SECTION").Value;

if (!checkBox1.Items.Conteins(section))

checkBox1.Items.Add(section);

string name = el1.Attributes.GetNamedItem("NAME").Value;

if (!checkBox2.Items.Conteins(name))

checkBox2.Items.Add(name);

string surname = el1.Attributes.GetNamedItem("SURNAME").Value;

if (!checkBox3.Items.Conteins(surname))

checkBox3.Items.Add(surname);

string faculty = el1.Attributes.GetNamedItem("FACULTY").Value;

if (!checkBox1.Items.Conteins(faculty))

checkBox1.Items.Add(faculty);

string schedule = el1.Attributes.GetNamedItem("SCUDULE").Value;

if (!checkBox1.Items.Conteins(schedule))

checkBox1.Items.Add(schedule);

string competition = el1.Attributes.GetNamedItem("COMPETITIONS").Value;

if (!checkBox1.Items.Conteins(competition))

checkBox1.Items.Add(competition);

string visitor = el1.ParentNode.Attributes.GetNamedItem("VISITOR").Value;

if (!checkBox1.Items.Conteins(visitor))

checkBox1.Items.Add(visitor);

checkBox2.Items.Add(el1.Attributes.GetNamedItem("NAME").Value);

checkBox3.Items.Add(el1.Attributes.GetNamedItem("SURNAME").Value);

checkBox4.Items.Add(el1.Attributes.GetNamedItem("FACULTY").Value);

checkBox5.Items.Add(el1.Attributes.GetNamedItem("SCUDULE").Value);

checkBox6.Items.Add(el1.Attributes.GetNamedItem("COMPETITIONS").Value);

checkBox7.Items.Add(el1.ParentNode.Attributes.GetNamedItem("VISITOR").Value);

}

}

}

}

public static Sportsman Info(XmlNode node)

{

Sportsman nw = new Sportsman();

nw.Section = node.ParentNode.ParentNode.Attributes.GetNamedItem("SECTION").Value;

nw.Name = node.Attributes.GetNamedItem("NAME").Value;

nw.Surname = node.Attributes.GetNamedItem("SURNAME").Value;

nw.Faculty = node.Attributes.GetNamedItem("FACULTY").Value;

nw.Schedule = node.Attributes.GetNamedItem("SCHEDULE").Value;

nw.Competition = node.Attributes.GetNamedItem("COMPETITION").Value;

nw.Visitor = node.ParentNode.Attributes.GetNamedItem("VISITOR").Value;

return nw;

}

\*/

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

richTextBox1.Clear();

Sportsman \_sportsman = OurSportsman();

if (radioButton1.Checked)

{

IStrategy CurrentStrategy = new Linq(path);

result = CurrentStrategy.Algorithm(\_sportsman, path);

Output(result);

}

if (radioButton2.Checked)

{

IStrategy CurrentStrategy = new Dom(path);

result = CurrentStrategy.Algorithm(\_sportsman, path);

Output(result);

}

if (radioButton3.Checked)

{

IStrategy CurrentStrategy = new Sax(path);

result = CurrentStrategy.Algorithm(\_sportsman, path);

Output(result);

}

}

private void button2\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

richTextBox1.Clear();

checkBox1.Checked = false;

checkBox2.Checked = false;

checkBox3.Checked = false;

checkBox4.Checked = false;

checkBox5.Checked = false;

checkBox6.Checked = false;

checkBox7.Checked = false;

checkBox1.Text = null;

checkBox2.Text = null;

checkBox3.Text = null;

checkBox4.Text = null;

checkBox5.Text = null;

checkBox6.Text = null;

checkBox7.Text = null;

radioButton1.Checked = false;

radioButton2.Checked = false;

radioButton3.Checked = false;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

class Linq : IStrategy

{

List<Sportsman> info = new List<Sportsman>();

XDocument doc = new XDocument();

public Linq(string path)

{

doc = XDocument.Load(path);

}

public List<Sportsman> Algorithm(Sportsman sportsman, string path)

{

List<XElement> match = (from val in doc.Descendants("sportsman")

where

((sportsman.Section == null || sportsman.Section == val.Parent.Parent.Atribute("SECTION").Value) &&

(sportsman.Name == null || sportsman.Name == val.Atribute("NAME").Value) &&

(sportsman.Surname == null || sportsman.Surname == val.Atribute("SURNAME").Value) &&

(sportsman.Faculty == null || sportsman.Faculty == val.Atribute("FACULTY").Value) &&

(sportsman.Schedule == null || sportsman.Schedule == val.Atribute("SCHEDULE").Value) &&

(sportsman.Competition == null || sportsman.Competition == val.Atribute("COMPETITION").Value) &&

(sportsman.Visitor == null || sportsman.Visitor == val.Parent.Atribute("VISITOR").Value))

select val).ToList();

foreach (XElement obj in match)

{

Sportsman sportsman1 = new Sportsman();

sportsman1.Section = obj.Parent.Parent.Atribute("SECTION").Value;

sportsman1.Name = obj.Atribute("NAME").Value;

sportsman1.Surname = obj.Atribute("SURNAME").Value;

sportsman1.Faculty = obj.Atribute("FACULTY").Value;

sportsman1.Schedule = obj.Atribute("SCHEDULE").Value;

sportsman1.Competition = obj.Atribute("COMPETITION").Value;

sportsman1.Visitor = obj.Atribute("VISITOR").Value;

}

return info;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml;

namespace Lab2

{

class Dom : IStrategy

{

XmlDocument doc = new XmlDocument();

public Dom(string path)

{

doc.Load(path);

}

public List<Sportsman> Algorithm(Sportsman sportsman, string path)

{

List<List<Sportsman>> info = new List<List<Sportsman>>();

try

{

if (sportsman.Section != null) info.Add(SearchByParam("section", "SECTION", sportsman.Section, doc, 0));

if (sportsman.Name != null) info.Add(SearchByParam("name", "NAME", sportsman.Name, doc, 2));

if (sportsman.Surname != null) info.Add(SearchByParam("surname", "SURNAME", sportsman.Surname, doc, 2));

if (sportsman.Faculty != null) info.Add(SearchByParam("faculty", "FACULTY", sportsman.Faculty, doc, 2));

if (sportsman.Schedule != null) info.Add(SearchByParam("schedule", "SCHEDULE", sportsman.Schedule, doc, 2));

if (sportsman.Competition != null) info.Add(SearchByParam("competition", "COMPETITION", sportsman.Competition, doc, 2));

if (sportsman.Visitor != null) info.Add(SearchByParam("visitor", "VISITOR", sportsman.Visitor, doc, 1));

}

catch { }

return Cross(info);

}

private List<Sportsman> SearchByParam(string v1, string v2, (object Name, XmlDocument doc, int) p)

{

throw new NotImplementedException();

}

private List<Sportsman> Cross(List<List<Sportsman>> info)

{

throw new NotImplementedException();

}

private List<Sportsman> SearchByParam(string v1, string v2, object section, XmlDocument doc, int v3)

{

throw new NotImplementedException();

}

}

}