**Министерство образования Республики Беларусь**

## Учреждение образования Международный государственный экологический университет имени А. Д. Сахарова

Факультет мониторинга окружающей среды

Кафедра экологических информационных систем

## РАЗРАБОТКА СТРУКТУРЫ ДАННЫХ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ХРАНЕНИЯ И ОТОБРАЖЕНИЯ ДАННЫХ О РЕЗУЛЬТАТАХ ПРОФИЛАКТИЧЕСКИХ ОСМОТРОВ

\_\_\_\_\_\_\_\_\_\_\_\_В.В. Шундрик

\_\_\_\_\_\_\_\_\_\_\_\_И.В. Лефанова

Минск 2021

**Описание**

Необходимо создать структуру данных. Разработать ПО которое будет хранить и отображать информацию о результатах профилактических осмотров.

**Структура класса**

Для создания структуры данных создаём класс Person, который будет хранить информацию о проф.осмотрах.

public class Person

{

public int nomer;

public String FIO;

public String Sex;

private DateTime bd;

public int growth;

public int weight;

public String nevrolog;

public String lor;

public String okylist;

public String xiryrg;

public String derma;

public String pcix;

public String stomac;

public String terapevt;

Поле nomer хранит номер осмотра, поле FIO хранит ФИО пациента, поле Sex хранит информацию о поле пациента, поле bd хранит дату рождения пациента, growth – рост пациента, weight – массу тела пациента, nevrolog – диагноз невролога, lor – диагноз ЛОРа, okylist – диагноз окулиста, xiryrg – диагноз хирурга, derma – диагноз дерматолога, pcix – диагноз психиатра, stomac – диагноз стоматолога, terapevt – диагноз терапевта .

**Формы**

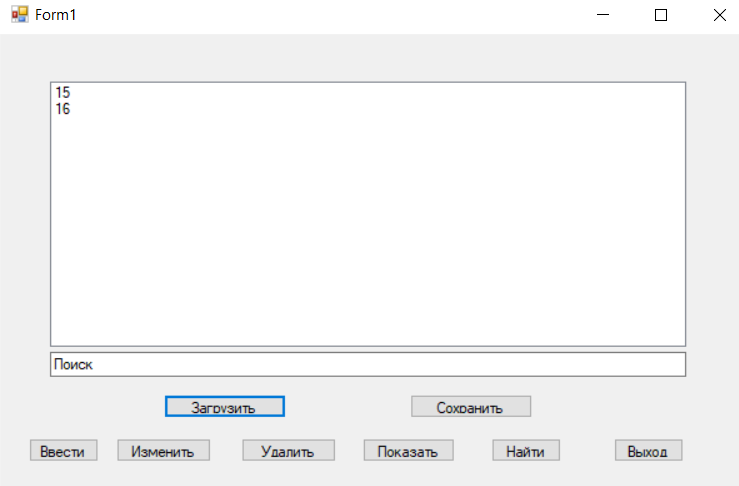


Рисунок 1 – главная форма

В программе есть кнопки, которые позволяют показать информацию выбранного осмотра (Рисунок 2). Есть возможность добавления новой записи с информацией о осмотре (Рисунок 3). В программе присутствуют кнопки для изменения (Рисунок 4) и удаления выбранной записи. Есть кнопка для поиска номера осмотра по ФИО пациента (Рисунок 5). Реализована возможность загрузки данных из текстового файла и сохранения в текстовый файл. Также есть кнопка, осуществляющая выход из программы.

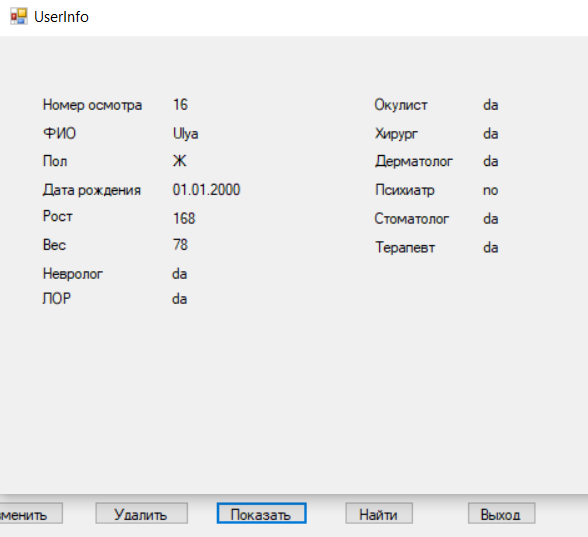


Рисунок 2 – отображение информации о осмотре

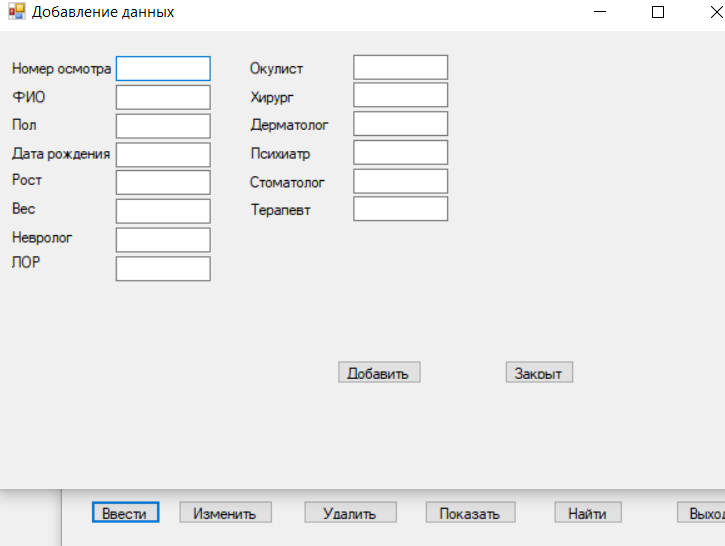


Рисунок 3 – добавление новой записи

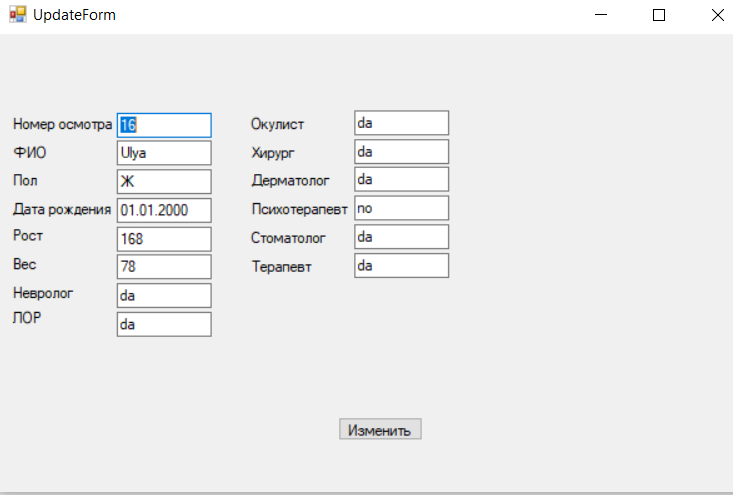
****

Рисунок 4 – изменение уже существующей записи

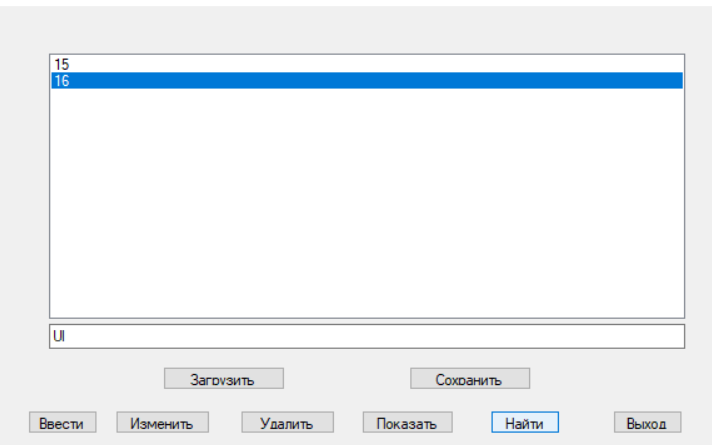
****

Рисунок 5 – поиск записи по ФИО

**Хранение данных**

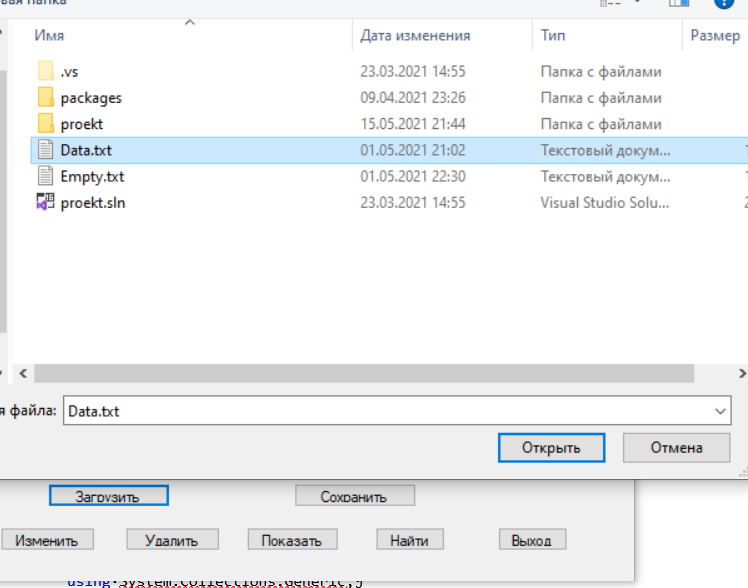
Информация о проф. осмотрах может загружаться и сохраняться в текстовых файлах.

Рисунок 6 – загрузка текстового файла с информацией

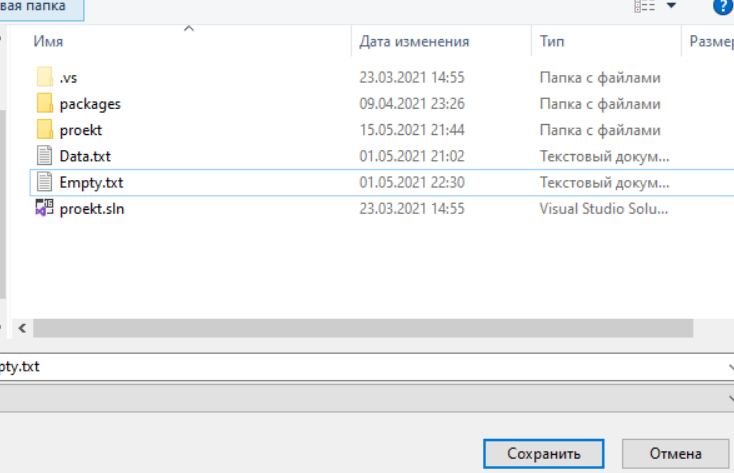


Рисунок 7 – сохранение в текстовый файл

**Код программы**

using OpenXmlPowerTools; // класс Person

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace proekt

{

public class Person

{

public int nomer;

public String FIO;

public String Sex;

private DateTime bd;

public int growth;

public int weight;

public String nevrolog;

public String lor;

public String okylist;

public String xiryrg;

public String derma;

public String pcix;

public String stomac;

public String terapevt;

public Person(){}

public Person(int nom, string name, string sex, DateTime born, int grow, int weig, string nevr, string LOR, string okyl,

string xir, string derm, string pci, string stom, string ter)

{

nomer = nom; FIO = name; Sex = sex;

bd = born; growth = grow; weight = weig;

nevrolog = nevr; lor = LOR; okylist = okyl;

xiryrg = xir; derma = derm;

pcix = pci; stomac = stom; terapevt = ter;

}

public Person(Person p)

{

nomer = p.nomer; FIO = p.FIO; Sex = p.Sex;

bd = p.bd; growth = p.growth; weight = p.weight;

nevrolog = p.nevrolog; lor = p.lor; okylist = p.okylist;

xiryrg = p.xiryrg; derma = p.derma;

pcix = p.pcix; stomac = p.stomac; terapevt = p.terapevt;

}

public String to\_ListItem()

{ return this.FIO + " " + bd.ToString("dd/MM/yyyy"); }

public int getNomer()

{

return nomer;

}

public void setNomer(int nomer)

{

this.nomer = nomer;

}

public string getFIO()

{

return FIO;

}

public void setFIO(string FIO)

{

this.FIO = FIO;

}

public string getSex()

{

return Sex;

}

public void setSex(string sex)

{

this.Sex = sex;

}

public DateTime getBd()

{

return bd;

}

public void setBd(DateTime bd)

{

this.bd = bd;

}

public int getGrougth()

{

return growth;

}

public void setGrougth(int grougth)

{

this.growth = grougth;

}

public int getWeigth()

{

return weight;

}

public void setWeigth(int weigth)

{

this.weight = weight;

}

public string getNevrolog()

{

return nevrolog;

}

public void setNevrolog(string nevrolog)

{

this.nevrolog = nevrolog;

}

public string getLor()

{

return lor;

}

public void setLor(string lor)

{

this.lor = lor;

}

public string getOkylist()

{

return okylist;

}

public void setOkylist(string okylist)

{

this.okylist = okylist;

}

public string getXiryrg()

{

return xiryrg;

}

public void setXiryrg(string xiryrg)

{

this.xiryrg = xiryrg;

}

public string getDerma()

{

return derma;

}

public void setDerma(string derma)

{

this.derma = derma;

}

public string getPcix()

{

return pcix;

}

public void setPcix(string pcix)

{

this.pcix = pcix;

}

public string getStomac()

{

return stomac;

}

public void setstomac(string stomac)

{

this.stomac = stomac;

}

public string getTerapevt()

{

return terapevt;

}

public void setTerapevt(string terapevt)

{

this.terapevt = terapevt;

}

}

//главная форма

using System;

using System.Collections.Generic;

using System.Collections;

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 proekt

{

public partial class Form1 : Form

{

dobavka db = new dobavka();

private List<Person> listPerson = new List<Person>();

private string fileName;

public Form1()

{

InitializeComponent();

}

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

{

listBox1.Items.Clear();

db = new dobavka(listPerson);

db.ShowDialog();

string[] str = null;

int i = 0;

foreach (Person p in listPerson)

{

string[] str1 = { System.Convert.ToString(p.nomer) };

str = str1;

listBox1.Items.AddRange(str);

}

}

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

{

if (listBox1.SelectedIndex >= 0)

{

listPerson.RemoveAt(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(listBox1.SelectedIndex);

}

}

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

{

this.Close();

}

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

{

string str = listBox1.SelectedItem.ToString();

foreach (Person person in listPerson)

{

if (person.getNomer().ToString().Equals(str))

{

UpdateForm change = new UpdateForm(person);

change.ShowDialog();

Person p = change.getPerson();

listPerson.Remove(person);

listPerson.Add(p);

listBox1.Items.Remove(listBox1.SelectedItem);

listBox1.Items.Add(p.getNomer());

return;

}

}

}

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

{

string searchText = textBox1.Text;

foreach (Person person in listPerson)

{

if (person.getFIO().Contains(searchText))

{

var items = listBox1.Items;

int counter = 0;

foreach (var item in items)

{

if (person.getNomer().ToString().Equals(item.ToString()))

{

listBox1.SetSelected(counter, true);

return;

}

counter++;

}

}

}

}

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

{

var selectedItem = listBox1.SelectedItem;

foreach (Person person in listPerson)

{

if (person.getNomer().ToString().Equals(selectedItem.ToString()))

{

UserInfo userInfo = new UserInfo(person);

userInfo.setInfo(person);

userInfo.ShowDialog();

}

}

}

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

{

OpenFileDialog fileDialog = new OpenFileDialog();

fileDialog.ShowDialog();

fileName = fileDialog.FileName;

string file = File.ReadAllText(fileName).Trim();

String[] rows = file.Split('\n');

foreach (string values in rows)

{

String[] numbers = values.Split(' ');

Person value = new Person(

Convert.ToInt32(numbers[0]), numbers[1],

numbers[2], Convert.ToDateTime(numbers[3]),

Convert.ToInt32(numbers[4]), Convert.ToInt32(numbers[5]),

numbers[6], numbers[7],

numbers[8], numbers[9],

numbers[10], numbers[11],

numbers[12], numbers[13]);

foreach (Person p in listPerson)

{

if (p.getNomer().Equals(value.getNomer()))

{

ThrErrorBox("Введенный номер записи уже существует");

return;

}

}

listPerson.Add(value);

listBox1.Items.Add(value.getNomer().ToString());

}

}

private DialogResult ThrErrorBox()

{

DialogResult result = MessageBox.Show(

"Проверьте введенные данные",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

this.TopMost = true;

return result;

}

private DialogResult ThrErrorBox(string message)

{

DialogResult result = MessageBox.Show(

message,

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

this.TopMost = true;

return result;

}

private void button4\_Click(object sender, EventArgs e)

{

string text = "";

foreach (Person value in listPerson)

{

text += value.getNomer() + " " + value.getFIO() + " " + value.getSex() + " " + value.getBd().ToString("dd.MM.yyyy")

+ " " + value.getGrougth() + " " + value.getWeigth()

+ " " + value.getNevrolog() + " " + value.getLor() + " " + value.getOkylist()

+ " " + value.getXiryrg() + " " + value.getDerma()

+ " " + value.getPcix() + " " + value.getStomac() + " " + value.getTerapevt() + "\n";

}

SaveFileDialog fileDialog = new SaveFileDialog();

fileDialog.ShowDialog();

fileName = fileDialog.FileName;

File.WriteAllText(fileName, text);

}

}

}

//форма с выводом информации о осмотре

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 proekt

{

public partial class UserInfo : Form

{

Person person;

public UserInfo()

{

InitializeComponent();

}

public UserInfo(Person person)

{

InitializeComponent();

this.person = person;

}

public void setInfo(Person person)

{

label15.Text = person.getNomer().ToString();

label16.Text = person.getFIO().ToString();

label17.Text = person.getSex().ToString();

label18.Text = person.getBd().ToString("dd.MM.yyyy");

label19.Text = person.getGrougth().ToString();

label20.Text = person.getWeigth().ToString();

label21.Text = person.getNevrolog().ToString();

label22.Text = person.getLor().ToString();

label23.Text = person.getOkylist().ToString();

label24.Text = person.getXiryrg().ToString();

label25.Text = person.getDerma().ToString();

label26.Text = person.getPcix().ToString();

label27.Text = person.getStomac().ToString();

label28.Text = person.getTerapevt().ToString();

}

}

}

//форма для добавления новых записей

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace proekt

{

public partial class dobavka : Form

{

List<Person> persons = new List<Person>();

public dobavka()

{

InitializeComponent();

}

public dobavka(List<Person> persons)

{

InitializeComponent();

this.persons = persons;

}

public dobavka(Person person)

{

InitializeComponent();

this.persons.Add(person);

}

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

{

try {

DateTime.Parse(this.BDate.Text);

Person person = new Person(

int.Parse(this.nomer.Text),

this.FIO.Text,

this.Sex.Text,

DateTime.Parse(this.BDate.Text),

int.Parse(this.growth.Text),

int.Parse(this.weight.Text),

this.nevrolog.Text,

this.lor.Text,

this.okylist.Text,

this.xiryrg.Text,

this.derma.Text,

this.pcix.Text,

this.stomac.Text,

this.terapevt.Text);

if (person.getBd().CompareTo(DateTime.Now) > 0)

{

ThrErrorBox("Неверная дата, повторите попытку");

return;

}

foreach (Person p in persons)

{

if (person.getNomer().Equals(p.getNomer()))

{

ThrErrorBox("Номер уже существует, повторите попытку");

return;

}

}

persons.Add(person);

cleanPanel();

}

catch (Exception ex) {

ThrErrorBox();

}

}

private DialogResult ThrErrorBox()

{

DialogResult result = MessageBox.Show(

"Проверьте введенные данные",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

this.TopMost = true;

return result;

}

private DialogResult ThrErrorBox(string message)

{

DialogResult result = MessageBox.Show(

message,

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

this.TopMost = true;

return result;

}

private void cleanPanel() {

this.nomer.Text = "";

this.FIO.Text = "";

this.Sex.Text = "";

this.BDate.Text = "";

this.growth.Text = "";

this.weight.Text = "";

this.nevrolog.Text = "";

this.lor.Text = "";

this.okylist.Text = "";

this.xiryrg.Text = "";

this.derma.Text = "";

this.pcix.Text = "";

this.stomac.Text = "";

this.terapevt.Text = "";

}

public List<Person> getPersons(){

return persons;

}

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

{

this.Close();

}

}

}

// форма для изменения записей

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 proekt

{

public partial class UpdateForm : Form

{

private Person person;

public UpdateForm()

{

InitializeComponent();

}

public UpdateForm(Person person)

{

InitializeComponent();

this.person = person;

setValues();

}

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

{

try

{

DateTime.Parse(this.BDate.Text);

Person person = new Person(

int.Parse(this.nomer.Text),

this.FIO.Text,

this.Sex.Text,

DateTime.Parse(this.BDate.Text),

int.Parse(this.growth.Text),

int.Parse(this.weight.Text),

this.nevrolog.Text,

this.lor.Text,

this.okylist.Text,

this.xiryrg.Text,

this.derma.Text,

this.pcix.Text,

this.stomac.Text,

this.terapevt.Text);

if (person.getBd().CompareTo(DateTime.Now) > 0)

{

ThrErrorBox("Неверная дата, повторите попытку");

return;

}

this.person = person;

this.Close();

}

catch (Exception ex)

{

ThrErrorBox();

}

}

private DialogResult ThrErrorBox()

{

DialogResult result = MessageBox.Show(

"Проверьте введенные данные",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

this.TopMost = true;

return result;

}

private DialogResult ThrErrorBox(string message)

{

DialogResult result = MessageBox.Show(

message,

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Error,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

this.TopMost = true;

return result;

}

private void setValues() {

nomer.Text = person.getNomer().ToString();

FIO.Text = person.getFIO().ToString();

Sex.Text = person.getSex().ToString();

BDate.Text = person.getBd().ToString("dd.MM.yyyy");

growth.Text = person.getGrougth().ToString();

weight.Text = person.getWeigth().ToString();

nevrolog.Text = person.getNevrolog().ToString();

lor.Text = person.getLor().ToString();

okylist.Text = person.getOkylist().ToString();

xiryrg.Text = person.getXiryrg().ToString();

derma.Text = person.getDerma().ToString();

pcix.Text = person.getPcix().ToString();

stomac.Text = person.getStomac().ToString();

terapevt.Text = person.getTerapevt().ToString();

}

public Person getPerson()

{

return person;

}

}

}

**Заключение**

В данной программе реализована возможность хранения, отображения, изменения и удаления информации о результатах проф. осмотров. Для этого использовались возможности языка C# и Visual Studio