Лабораторная работа №6

Экономическая система

Вариант №15

*Постановка задачи*: Разрабатываемое вами приложение – это прообраз реальной экономической

информационной системы. Приложения такого уровня помогают бизнесу существовать,

оптимизировать свою работу и управление.

*Входные данные*:

* строка;

*Выходные данные*:

* massivBezTochki – преобразованный массив;

*Рабочие переменные*:

* i – счетчик цикла;
* element;
* massivBezTochki.
* OneElement.
* Arr.

*Описание стандартных функций и библиотек*

* Console.WriteLine() – вывод данные на экран;
* ToCharArray() – копирует знаки в массив чар;
* Ref – передает ссылку на ячейку памяти в куче
* Out – передает ссылку на ячейку памяти в куче, объявление переменной в методе.
* Length – длинна массива
* Console.ReadLine() – считывание данные с клавиатуры;

*Классы и процедуры, созданные автором*

* Flat - Класс квартира
* Flats - Класс квартиры
* Client - Класс клиент
* Clients - Класс клиенты
* Types - Класс типы

*Тестовый пример*

Если пользователь введёт 5 произвольных слов то программа отобразит <<Массив введенных слов, далее программа удалит слова идентичные первому введенному слову и те слова в которых первая буква встречается еще раз. >>.

*Текст программы:*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using DB\_RieltorCompany;

using System.Text.RegularExpressions;

//при удалении клиентов потом мдёт ошибка - она не может его найти при заполнении формы....

namespace DB\_Winform

{

public partial class Form1 : Form

{

Clients myClients;

Flats myFlats;

types myTypes;

private Form2 frmAdd;

int currentFlatNumber;

public Form1()

{

InitializeComponent();

}

private void FillForm(Flat f)//заполнение главной формы

{

this.Text = "Недвижимость номер:" + f.ID;

cmbRealtyTypeAddNedvObject.Text = f.RealtyType;//заполнение окон формы

txtBoxAddPriceNedvObject.Text = f.Price.ToString();//заполнение окон формы

cmbClientAddNedvObject.Text = f.clienttt.Name;

cmbFloorAddNedvObject.Text = f.Floor.ToString();

txtBoxAreaAddNedvOjbect.Text = f.Area.ToString();

txtBoxRoomsCountAddNedvObject.Text = f.RoomsCount.ToString();

txtBoxOpisanieAddNedvObject.Text = f.Description;

txtBoxAdressAddNedvObject.Text = f.Address;

cmbLikved.Text = myTypes.getTypes(f.Likvid-1).ToString();

}

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

{

Form1\_Resize(sender, e);

myClients=new Clients("clients.csv");

myFlats= new Flats("data.csv", myClients);

myTypes = new types("types.csv");

cmbRealtyTypeAddNedvObject.Items.Add("Квартира");

cmbRealtyTypeAddNedvObject.Items.Add("Дом");

cmbRealtyTypeAddNedvObject.Items.Add("Комната");

cmbRealtyTypeAddNedvObject.Items.Add("Земля");

for (int i = 0; i < myClients.clientsCount; i++)

{

cmbClientAddNedvObject.Text = myClients.getClient(0).Name;

chkbxClients.Items.Add(myClients.getClient(i));

}

for (int i = 0; i < 5; i++)

{

cmbFloorAddNedvObject.Items.Add(i);

}

for (int i = 0; i < myTypes.typesCount; i++)

{

cmbLikved.Items.Add(myTypes.getTypes(i).Likved);

}

if (myFlats.flatsCount > 0)//?????????????????????

{

this.currentFlatNumber = 0;

FillForm(myFlats.getFlat(currentFlatNumber));

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

{

chkbxClients.SetItemCheckState(i, CheckState.Unchecked);

}

}

}

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

{

btnNextRealty.Left = this.Width - btnNextRealty.Width-30;

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (MessageBox.Show("Вы уверены?", "Вопрос", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

myFlats.saveFlatsToFile();

myClients.saveClientsToFile();

e.Cancel = false;

}

else

{

e.Cancel = true;

}

}

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

{

if (myFlats.flatsCount == 0)

{

MessageBox.Show(

"Переход не может быть выполнен!",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1);

}

else

{

currentFlatNumber -= 1;

if (currentFlatNumber < 0) { currentFlatNumber = myFlats.flatsCount - 1; }

FillForm(myFlats.getFlat(currentFlatNumber));

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

{

chkbxClients.SetItemCheckState(i, CheckState.Unchecked);

}

}

}

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

{

if (myFlats.flatsCount == 0)

{

MessageBox.Show(

"Переход не может быть выполнен!",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1);

}

else

{

currentFlatNumber += 1;

if (currentFlatNumber > myFlats.flatsCount - 1)

{

currentFlatNumber = 0;

}

FillForm(myFlats.getFlat(currentFlatNumber));

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

{

chkbxClients.SetItemCheckState(i, CheckState.Unchecked);

//if (chkbxClients == myClients.getClient(i).ToString())

//{

// chkbxClients.SetItemCheckState(i , CheckState.Checked);

// break;

//}

}

}

}

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

{

if (myFlats.flatsCount == 0)

{

MessageBox.Show(

"Тут нечего изменять!",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1);

txtBoxAreaAddNedvOjbect.Clear();

cmbFloorAddNedvObject.ResetText();

txtBoxAddPriceNedvObject.Clear();

txtBoxOpisanieAddNedvObject.Clear();

txtBoxAdressAddNedvObject.Clear();

txtBoxRoomsCountAddNedvObject.Clear();

cmbRealtyTypeAddNedvObject.ResetText();

}

else

{

for (int i = 0; i < myClients.clientsCount; i++)

{

if (chkbxClients.GetItemChecked(i))

{

myFlats.changeClientOnFlat(currentFlatNumber, myClients.getClient(i));

myFlats.getFlat(currentFlatNumber).ClientID = myClients.getClient(i).ID;

cmbClientAddNedvObject.Text = myClients.getClient(i).Name;

break;

if (i == myClients.clientsCount - 1)

{

myFlats.changeClientOnFlat(currentFlatNumber, myFlats.getFlat(currentFlatNumber).clienttt);

myFlats.getFlat(currentFlatNumber).ClientID = myFlats.getFlat(currentFlatNumber).clienttt.ID;

}

}

}

bool allIsGood;

errorProvider1.Clear();

//string patternAdress = @"^[а-яА-ЯёЁa-zA-Z0-9]+$";

//string patternAdressletters = @"^[а-яА-ЯёЁa-zA-Z]+$";

//string patternAdressnumbers = @"^[0 - 9] +$";

if (isNumberINString(txtBoxAddPriceNedvObject.Text) == false) // обработка ошибок ЦЕНА

{

errorProvider1.SetError(txtBoxAddPriceNedvObject, "Не указана цена");

allIsGood = false;

}

/////////////

else if (isDigitsInString(txtBoxAdressAddNedvObject.Text) == false || isNumberINString(txtBoxAdressAddNedvObject.Text) == true)//обработка ошибок АДРЕСС

{

errorProvider1.SetError(txtBoxAdressAddNedvObject, "Тут должны быть цифры и буквы");

allIsGood = false;

}

else if (isNumberINString(txtBoxAreaAddNedvOjbect.Text) == false)// обработка ошибок ПЛОЩАДЬ

{

errorProvider1.SetError(txtBoxAreaAddNedvOjbect, "Не верные данные");

allIsGood = false;

}

////////////////

else if (String.IsNullOrWhiteSpace(txtBoxAdressAddNedvObject.Text))// обработка ошибок АДРЕСС

{

errorProvider1.SetError(txtBoxAdressAddNedvObject, "Не указан адресс");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(cmbFloorAddNedvObject.Text))// обработка ошибок ЭТАЖ

{

errorProvider1.SetError(cmbFloorAddNedvObject, "Не указан этаж");

allIsGood = false;

}

else if (isNumberINString(cmbFloorAddNedvObject.Text) == false)// обработка ошибок ЭТАЖ

{

errorProvider1.SetError(cmbFloorAddNedvObject, "Тут должны быть цифры");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(txtBoxRoomsCountAddNedvObject.Text))// обработка ошибок КОЛИЧЕСТВО КОМНАТ

{

errorProvider1.SetError(txtBoxRoomsCountAddNedvObject, "Пустое поле");

allIsGood = false;

}

else if (isNumberINString(txtBoxRoomsCountAddNedvObject.Text) == false)// обработка ошибок КОЛИЧЕСТВО КОМНАТ

{

errorProvider1.SetError(txtBoxRoomsCountAddNedvObject, "Тут должны быть цифры");

allIsGood = false;

}

//else if (isNumberINString(cmbClientAddNedvObject.Text) == false)// обработка ошибок КЛИЕНТ

//{

// errorProvider1.SetError(cmbClientAddNedvObject, "Здесь должен быть номер клиента");

// allIsGood = false;

//}

//else if (String.IsNullOrWhiteSpace(cmbClientAddNedvObject.Text))// обработка ошибок КЛИЕНТ

//{

// errorProvider1.SetError(cmbClientAddNedvObject, "Не указан клиент");

// allIsGood = false;

//}

//else if (searchClient(cmbClientAddNedvObject.Text) == false)// обработка ошибок КЛИЕНТ

//{

// errorProvider1.SetError(cmbClientAddNedvObject, "Такой клиент не найден");

// allIsGood = false;

//}

else if (String.IsNullOrWhiteSpace(cmbRealtyTypeAddNedvObject.Text))// обработка ошибок ТИП НЕДВИЖИМОСТИ

{

errorProvider1.SetError(cmbRealtyTypeAddNedvObject, "Не указан тип недвижимости");

allIsGood = false;

}

else if (serchNedvijimost(cmbRealtyTypeAddNedvObject.Text) == false)// обработка ошибок ТИП НЕДВИЖИМОСТИ

{

errorProvider1.SetError(cmbRealtyTypeAddNedvObject, "Такого типа недвижимости не существует");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(txtBoxAreaAddNedvOjbect.Text))// обработка ошибок ПЛОЩАДЬ

{

errorProvider1.SetError(txtBoxAreaAddNedvOjbect, "Не указана площадь");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(txtBoxOpisanieAddNedvObject.Text))// обработка ошибок ОПИСАНИЕ

{

errorProvider1.SetError(txtBoxOpisanieAddNedvObject, "Не указано описание");

allIsGood = false;

}

else if (isNumberINString(cmbLikved.Text) == true)

{

errorProvider1.SetError(cmbLikved, "Не верные данные");

allIsGood = false;

}

else if (getLikvid(cmbLikved.Text) == 1)

{

errorProvider1.SetError(cmbLikved, "Такой ликвидности нет");

allIsGood = false;

}

else

{

errorProvider1.Clear();

allIsGood = true;

}

///////////////////

if (allIsGood == true)

{

myFlats.getFlat(currentFlatNumber).RoomsCount = Convert.ToInt32(txtBoxRoomsCountAddNedvObject.Text);

myFlats.getFlat(currentFlatNumber).Area = Convert.ToInt32(txtBoxAreaAddNedvOjbect.Text);

myFlats.getFlat(currentFlatNumber).Price = Convert.ToInt32(txtBoxAddPriceNedvObject.Text);

myFlats.getFlat(currentFlatNumber).Description = txtBoxOpisanieAddNedvObject.Text;

myFlats.getFlat(currentFlatNumber).Address = txtBoxAdressAddNedvObject.Text;

myFlats.getFlat(currentFlatNumber).Floor = Convert.ToInt32(cmbFloorAddNedvObject.Text);

myFlats.getFlat(currentFlatNumber).RealtyType = cmbRealtyTypeAddNedvObject.Text;

myFlats.getFlat(currentFlatNumber).Likvid = getLikvid(cmbLikved.Text);

//myFlats.getFlat(currentNumber).ID = rbClients.First(x.Checked== true).Tag;

////////////////////////////myFlats.getFlat(currentFlatNumber).ClientID = chkbxClients.SelectedIndex;

myFlats.saveFlatsToFile();

//myClients.saveClientsToFile();

//string checkedRadioTag="";

//foreach (RadioButton rb in rbClients)

//{

// if (rb.Checked==true)

// {

// checkedRadioTag = rb.Tag.ToString(); ;

// }

//}

//MessageBox.Show("Test");

}

}

}

public int getLikvid(string likvid)

{

for (int i = 0; i < myTypes.typesCount; i++)

{

if (likvid == myTypes.getTypes(i).Likved)

{

return myTypes.getTypes(i).ID;

}

}

return 1;

}

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

{

string s = "";

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

{

s = s + chkbxClients.GetItemChecked(i).ToString() + "\n";

chkbxClients.SetItemCheckState(i, CheckState.Unchecked);

}

try

{

chkbxClients.SetItemCheckState(chkbxClients.SelectedIndex, CheckState.Checked);

}

catch

{

chkbxClients.SetItemCheckState(0, CheckState.Checked);

}

}

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

{

if (myFlats.flatsCount == 1)

{

MessageBox.Show(

"Тут нечего удалять!",

"Сообщение",

MessageBoxButtons.OK,

MessageBoxIcon.Information,

MessageBoxDefaultButton.Button1);

}

else

{

if (myFlats.flatsCount == 1)

{

this.Text = "";

cmbRealtyTypeAddNedvObject.Text = "";

cmbFloorAddNedvObject.Text = "";

cmbClientAddNedvObject.Text = "";

txtBoxAdressAddNedvObject.Text = "";

txtBoxAddPriceNedvObject.Text = "";

txtBoxOpisanieAddNedvObject.Text = "";

txtBoxRoomsCountAddNedvObject.Text = "";

txtBoxAreaAddNedvOjbect.Text = "";

//cmbType.SelectedItem = "";

//cmbFloor.SelectedItem = "";

//cmbDeleteNedvObject.Items.RemoveAt(i);

//cmbDeleteNedvObject.ResetText();

}

else

{

myFlats.remove(myFlats.getFlat(currentFlatNumber));

btnPreviousRealty\_Click(sender, e);

//FillForm(myFlats.getFlat(0));

}

}

}

public string returnClient(string clientName)

{

for (int i = 0; i < myClients.clientsCount; i++)

{

if (clientName == myClients.getClient(i).Name)

{

return myClients.getClient(i).ID.ToString();

break;

}

}

return "a";

}

public bool serchNedvijimost(string stroka) //поиск введенных в строку значений с объектами комбо бокса

{

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

{

if (stroka == cmbRealtyTypeAddNedvObject.Items[i].ToString())

{

return true;

}

}

return false;

}

public bool searchClient(string name)// поиск имеющихся клиентов по id

{

for (int i = 0; i < myClients.clientsCount; i++)

{

if (myClients.getClient(i).Name == cmbClientAddNedvObject.Text)

{

return true;

break;

}

}

return false;

}

//^[а-яА-ЯёЁa-zA-Z0-9]+$

/// <summary>

/// Определяет есть ли в строке хотя бы одна цифра

/// </summary>

/// <param name="s"></param>

/// <returns></returns>

private bool isDigitsInString(string s)

{

//почитать про регулярные выражения!!!!

//можно проверять адрес на наличие и букв и цифр....!!!

string checkAlphabet = "0123456789";

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

{

if (s.Contains(checkAlphabet[i])) {return true;}

}

return false;

}

private bool isNumberINString(String s)//проверка на трай парсе

{

Int32 z=0;

return Int32.TryParse(s,out z);

}

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

{

myFlats.saveFlatsToFile();

myClients.saveClientsToFile();

}

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

{

this.Visible = false;

if (frmAdd == null)

{ frmAdd = new Form2(myFlats, myClients, myTypes, this); }

frmAdd.Show();

}

//private void append(ref Flats flatsTable, Clients clients)//добавить строку

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using DB\_RieltorCompany;

namespace DB\_Winform

{

public partial class Form2 : Form

{

private Form1 mainForm;

Flats myFlats;

Clients myClients;

types myTypes;

public Form2(Flats myFlats, Clients myClients, types myTypes, Form1 form1)

{

this.myFlats = myFlats;

this.myClients = myClients;

this.myTypes = myTypes;

mainForm = form1;

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

this.Text = "Добавление недвижимости";

cmbRealtyTypeAddNedvObject.Items.Add("Квартира");

cmbRealtyTypeAddNedvObject.Items.Add("Дом");

cmbRealtyTypeAddNedvObject.Items.Add("Комната");

cmbRealtyTypeAddNedvObject.Items.Add("Земля");

for (int i = 0; i < myClients.clientsCount; i++)

{

cmbClientAddNedvObject.Items.Add(myClients.getClient(i).Name);

}

for (int i = 0; i < 5; i++)

{

cmbFloorAddNedvObject.Items.Add(i);

}

for (int i = 0; i < myTypes.typesCount; i++)

{

cmbLikved.Items.Add(myTypes.getTypes(i).Likved);

}

}

private void Form2\_FormClosing(object sender, FormClosingEventArgs e)

{

mainForm.Visible = true;

this.Hide();

e.Cancel = true;

}

private bool isDigitsInString(string s)

{

//почитать про регулярные выражения!!!!

//можно проверять адрес на наличие и букв и цифр....!!!

string checkAlphabet = "0123456789";

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

{

if (s.Contains(checkAlphabet[i])) { return true; }

}

return false;

}

private bool isNumberINString(String s)//проверка на трай парсе

{

Int32 z = 0;

return Int32.TryParse(s, out z);

}

public string returnClient(string clientName)

{

for (int i = 0; i < myClients.clientsCount; i++)

{

if (clientName == myClients.getClient(i).Name)

{

return myClients.getClient(i).ID.ToString();

break;

}

}

return "a";

}

public int returnTipe(string typeN)

{

for (int i = 0; i < myTypes.typesCount; i++)

{

if (typeN == myTypes.getTypes(i).Likved)

{

return myTypes.getTypes(i).ID;

}

}

return 1;

}

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

{

bool allIsGood;

errorProvider1.Clear();

//string patternAdress = @"^[а-яА-ЯёЁa-zA-Z0-9]+$";

//string patternAdressletters = @"^[а-яА-ЯёЁa-zA-Z]+$";

//string patternAdressnumbers = @"^[0 - 9] +$";

if (isNumberINString(txtBoxAddPriceNedvObject.Text) == false) // обработка ошибок ЦЕНА

{

errorProvider1.SetError(txtBoxAddPriceNedvObject, "Не указана цена");

allIsGood = false;

}

/////////////

else if (isDigitsInString(txtBoxAdressAddNedvObject.Text) == false || isNumberINString(txtBoxAdressAddNedvObject.Text) == true)//обработка ошибок АДРЕСС

{

errorProvider1.SetError(txtBoxAdressAddNedvObject, "Тут должны быть цифры и буквы");

allIsGood = false;

}

else if (isNumberINString(txtBoxAreaAddNedvOjbect.Text) == false)// обработка ошибок ПЛОЩАДЬ

{

errorProvider1.SetError(txtBoxAreaAddNedvOjbect, "Не верные данные");

allIsGood = false;

}

////////////////

else if (String.IsNullOrWhiteSpace(txtBoxAdressAddNedvObject.Text))// обработка ошибок АДРЕСС

{

errorProvider1.SetError(txtBoxAdressAddNedvObject, "Не указан адресс");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(cmbFloorAddNedvObject.Text))// обработка ошибок ЭТАЖ

{

errorProvider1.SetError(cmbFloorAddNedvObject, "Не указан этаж");

allIsGood = false;

}

else if (isNumberINString(cmbFloorAddNedvObject.Text) == false)// обработка ошибок ЭТАЖ

{

errorProvider1.SetError(cmbFloorAddNedvObject, "Тут должны быть цифры");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(txtBoxRoomsCountAddNedvObject.Text))// обработка ошибок КОЛИЧЕСТВО КОМНАТ

{

errorProvider1.SetError(txtBoxRoomsCountAddNedvObject, "Пустое поле");

allIsGood = false;

}

else if (isNumberINString(txtBoxRoomsCountAddNedvObject.Text) == false)// обработка ошибок КОЛИЧЕСТВО КОМНАТ

{

errorProvider1.SetError(txtBoxRoomsCountAddNedvObject, "Тут должны быть цифры");

allIsGood = false;

}

else if (isNumberINString(cmbClientAddNedvObject.Text) == true)// обработка ошибок КЛИЕНТ

{

errorProvider1.SetError(cmbClientAddNedvObject, "Здесь не может быть цифр");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(cmbClientAddNedvObject.Text))// обработка ошибок КЛИЕНТ

{

errorProvider1.SetError(cmbClientAddNedvObject, "Не указан клиент");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(cmbRealtyTypeAddNedvObject.Text))// обработка ошибок ТИП НЕДВИЖИМОСТИ

{

errorProvider1.SetError(cmbRealtyTypeAddNedvObject, "Не указан тип недвижимости");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(cmbLikved.Text))

{

errorProvider1.SetError(cmbLikved, "Не указана Ликвидность");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(txtBoxAreaAddNedvOjbect.Text))// обработка ошибок ПЛОЩАДЬ

{

errorProvider1.SetError(txtBoxAreaAddNedvOjbect, "Не указана площадь");

allIsGood = false;

}

else if (String.IsNullOrWhiteSpace(txtBoxOpisanieAddNedvObject.Text))// обработка ошибок ОПИСАНИЕ

{

errorProvider1.SetError(txtBoxOpisanieAddNedvObject, "Не указано описание");

allIsGood = false;

}

else

{

errorProvider1.Clear();

allIsGood = true;

}

//if (isDigitsInString(txtBoxOpisanieAddNedvObject.Text)==false)// обработка ошибок ОПИСАНИЕ

//{

// errorProvider1.SetError(txtBoxOpisanieAddNedvObject, "В описании должна быть хотя бы одна цифра!!!");

// allIsGood = false;

//}

if (allIsGood == true)//Добавление недвижимости

{

string newDataStringFor = myFlats.nextId + ";" + cmbRealtyTypeAddNedvObject.Text + ";" + txtBoxRoomsCountAddNedvObject.Text + ";" + cmbFloorAddNedvObject.Text + ";" + txtBoxAreaAddNedvOjbect.Text + ";" + txtBoxAddPriceNedvObject.Text + ";" + txtBoxAdressAddNedvObject.Text + ";" + txtBoxOpisanieAddNedvObject.Text + ";" + returnClient(cmbClientAddNedvObject.Text) + ";" + returnTipe(cmbLikved.Text);// последний элемент не подходит!!!!!!!!!!!!!!!!!!!!!

if (myFlats.flatsCount == 0)

{

newDataStringFor = "1" + ";" + cmbRealtyTypeAddNedvObject.Text + ";" + txtBoxRoomsCountAddNedvObject.Text + ";" + cmbFloorAddNedvObject.Text + ";" + txtBoxAreaAddNedvOjbect.Text + ";" + txtBoxAddPriceNedvObject.Text + ";" + txtBoxAdressAddNedvObject.Text + ";" + txtBoxOpisanieAddNedvObject.Text + ";" + returnClient(cmbClientAddNedvObject.Text);// последний элемент не подходит!!!!!!!!!!!!!!!!!!!!!

}

myFlats.add(new Flat(newDataStringFor, myClients));

cmbRealtyTypeAddNedvObject.SelectedIndex = -1;

txtBoxAddPriceNedvObject.ResetText();

cmbClientAddNedvObject.SelectedIndex = -1;

cmbFloorAddNedvObject.SelectedIndex = -1;

txtBoxAreaAddNedvOjbect.ResetText();

txtBoxRoomsCountAddNedvObject.ResetText();

txtBoxOpisanieAddNedvObject.ResetText();

txtBoxAdressAddNedvObject.ResetText();

cmbLikved.SelectedIndex = -1;

MessageBox.Show("Добавлено!");

myFlats.saveFlatsToFile();

}

}

}

}

*Результат выполнения программы:*

