Федеральное государственное бюджетное образовательное учреждение

«Сибирский государственный университет

телекоммуникаций и информатики»

кафедра прикладной математики и кибернетики

Расчетно-графическая работа по

дисциплине “Основы визуального программирования”

Вариант-4 (Пункты задания: 3,7,9,10,12)

Тема: «Видеотека»

Выполнил: студент 2 курса группы ИП-813

Бурдуковский Илья Алексан6дрович

Проверил: Старший преподаватель кафедры ПМиК

Милешко Антон Владимирович

Новосибирск 2020г

План

1. Текст задания
2. Описание работы приложения
3. Код программы
4. Работа программы

Текст задания

**Основная идея:**

Некий любитель кино обладает огромным собранием фильмов (на видеокассетах, компакт-дисках, DVD-дисках), а также большим количеством друзей, которые тоже любят кино и берут фильмы, частенько забывая их вовремя отдавать. Поскольку и фильмов, и друзей слишком много, то наш любитель кино уже запутался, кто что взял и когда он увидит свои фильмы снова. Помогите киноману, написав приложение, обслуживающее базу данных с информацией о фильмах (название, жанр, режиссер, киностудия, актеры, краткая аннотация, и т.д.), о пользователях видеотеки (ФИО, адрес) и о том, кто и когда взял какой фильм. В приложении необходимо реализовать следующие функции:

1. Ввод данных о новом фильме

2. Ввод данных о новом пользователе

3. Возможность отметить кто какой фильм взял

4. Получение сведений о том, какие фильмы брал данный пользователь с распечаткой

5. Получение сведений о том, кто и когда брал данный фильм с распечаткой

6. Получение списка должников с распечаткой

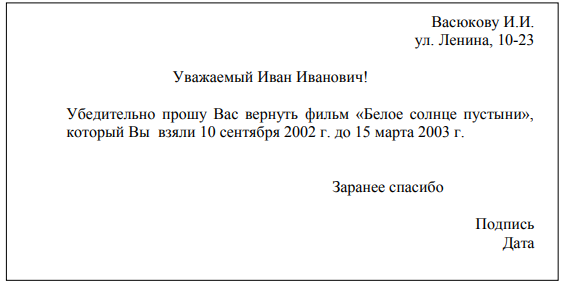
7. Получение списка отданных фильмов с распечаткой

8. Поиск фильма по названию (неточный поиск)

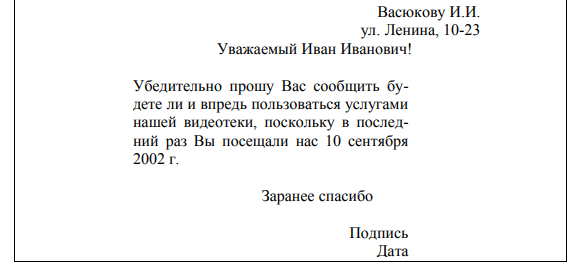
9. Поиск пользователя по ФИО (неточный поиск)

10. Поиск фильма по жанру и по актерам

11. Составление письма должнику



12. Составление письма давно не посещавшему видеотеку пользователю



Описание работы приложения

Для полноценной работы программы была создана обширная база данных.

Перед началом работы приложения, программа подключалась к источнику данных с помощью соединения OleDbConnection.

Считывания нужных данных из БД производилось за счет команд запросов SQL. Считанные данные из БД выводились в datagridview с помощью классов OleDbDataAdapter и DataSet. OleDbDataAdapter представлял набор команд данных и подключение к базе данных, которые использует для заполнения DataSet и обновления источника данных. DataSet представлял кэш данных памяти. В программе использовался метод Fill, который добавляет и обновляет строки в DataSet. Вся информация была получена с помощью пространства имен System.Data.OleDb. Данные могли вноситься в БД, так и считываться из нее.

После успешной компиляции программы, пользователю будет доступна форма с данными из таблиц «Фильмы» и «Пользователи», а также выбором различных пунктов, с помощью которых пользователь сможет взаимодействовать с программой. После выбора пункта меню, пользователю откроется дополнительная форма с информацией. Некоторые label’ы, присутствующие на формах, могут содержать информацию, которая была взята программой из базы данных.

Также в программе предоставлен поиск информации в БД. Реализован поиск с помощью команд запросов SQL.

Код программы

EntryForm

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.Data.OleDb;

namespace ргр

{

public partial class EntryForm : Form

{

OleDbConnection myConnnect;

OleDbDataAdapter da, da1;

DataSet ds, ds1;

public EntryForm()

{

InitializeComponent();

string connect = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:/personal/учеба/4 семестр/Визуалка/ргр/ргр/ргр/videoteka.mdb;";

myConnnect = new OleDbConnection(connect);

DataLoad();

}

public void DataLoad()

{

try

{

string sqlstring = "SELECT \* FROM [Users]";

ds = new DataSet();

da = new OleDbDataAdapter(sqlstring, myConnnect);

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

string sqlstring1 = "SELECT \* FROM [Films]";

ds1 = new DataSet();

da1 = new OleDbDataAdapter(sqlstring1, myConnnect);

da1.Fill(ds1);

dataGridView2.DataSource = ds1.Tables[0];

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

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

{

string sqlstring = "SELECT \* FROM [Users]";

archive AddRec = new archive();

AddRec.Owner = this;

AddRec.Show();

}

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

{

string Name =" ", Address = " ";

int i = dataGridView1.CurrentCell.RowIndex;

Name = dataGridView1.Rows[i].Cells[1].Value.ToString();

Address = dataGridView1.Rows[i].Cells[2].Value.ToString();

Letter AddRec = new Letter(Name,Address);

AddRec.Owner = this;

AddRec.ShowDialog();

}

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

{

MissingFilms AddRec = new MissingFilms();

AddRec.Owner = this;

AddRec.ShowDialog();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

string sqlstring = "SELECT \* FROM [Users] WHERE ФИО LIKE '" + "%" + textBox1.Text + "%" + "'";

ds = new DataSet();

da = new OleDbDataAdapter(sqlstring, myConnnect);

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

try

{

string sqlstring = "SELECT \* FROM [Films] WHERE Жанр LIKE '" + textBox2.Text + "%" + "' OR Актёры LIKE '" + "%" + textBox2.Text + "%" +"'";

ds = new DataSet();

da = new OleDbDataAdapter(sqlstring, myConnnect);

da.Fill(ds);

dataGridView2.DataSource = ds.Tables[0];

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

}

}

Archive

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.Data.OleDb;

namespace ргр

{

public partial class archive : Form

{

OleDbConnection myConnnect;

OleDbDataAdapter da, da1;

DataSet ds, ds1;

public int AddFilmID=0, AddUserID=0;

public archive()

{

InitializeComponent();

string connect = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:/personal/учеба/4 семестр/Визуалка/ргр/ргр/ргр/videoteka.mdb;";

myConnnect = new OleDbConnection(connect);

DataLoad();

}

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

{

DataLoad();

EntryForm Main = this.Owner as EntryForm;

Main.DataLoad();

}

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

{

AddDebtor AddRec = new AddDebtor(AddUserID,AddFilmID);

AddRec.Owner = this;

AddRec.ShowDialog();

if(AddUserID!=0 && AddFilmID != 0)

{

EntryForm ent = this.Owner as EntryForm;

for (int i = 0; i < ent.dataGridView2.RowCount; i++)

{

if ((ent.dataGridView2.Rows[i].Cells[0].Value.ToString() == Convert.ToString(AddFilmID)) && (ent.dataGridView2.Rows[i].Cells[6].Value.ToString() == "False"))

{

MessageBox.Show("Этот фильм отсутствует");

return;

}

}

DateTime thisDay = DateTime.Now;

string query = "INSERT INTO [Выдача фильмов] ([ID взявшего],[ID фильма],Дата,[Состояние возврата]) " +

"VALUES (" + AddUserID + "," + AddFilmID + ",'" + thisDay + "'," + false + ")";

ds = new DataSet();

da = new OleDbDataAdapter(query, myConnnect);

da.Fill(ds);

query = "UPDATE [Выдача фильмов], [Films] SET [Films].[Наличие] = False " +

"WHERE [Выдача фильмов].[Состояние возврата] = False AND [Films].Код = [Выдача Фильмов].[ID фильма]";

ds = new DataSet();

da = new OleDbDataAdapter(query, myConnnect);

da.Fill(ds);

MessageBox.Show("База обновлена");

AddUserID = 0;

AddFilmID = 0;

}

AddUserID = AddFilmID = 0;

DataLoad();

}

private void dataGridView2\_CellValueChanged(object sender, DataGridViewCellEventArgs e)

{

int i = dataGridView2.CurrentCell.RowIndex;

string FID = dataGridView2.Rows[i].Cells[2].Value.ToString();

if (dataGridView2.Rows[i].Cells[4].Value.ToString() == "True")

{

string query = "UPDATE [Выдача фильмов], [Films] " +

"SET [Выдача фильмов].[Состояние возврата] = True, [Films].[Наличие] = True " +

"WHERE [Films].[Код] = [Выдача Фильмов].[ID фильма] AND [Films].[Название] = '" + FID + "'";

ds = new DataSet();

da = new OleDbDataAdapter(query, myConnnect);

da.Fill(ds);

}

else

{

string query = "UPDATE [Выдача фильмов], [Films] " +

"SET [Выдача фильмов].[Состояние возврата] = False, [Films].[Наличие] = False " +

"WHERE [Films].[Код] = [Выдача Фильмов].[ID фильма] AND [Films].[Название] = '" + FID + "'";

ds = new DataSet();

da = new OleDbDataAdapter(query, myConnnect);

da.Fill(ds);

}

DataLoad();

}

private void DataLoad()

{

try

{

string query = "SELECT[Users].\* FROM[Users] INNER JOIN[Выдача Фильмов] ON [Users].Код = [Выдача Фильмов].[ID взявшего] " +

"WHERE [Выдача фильмов].[Состояние возврата] = False";

ds = new DataSet();

da = new OleDbDataAdapter(query, myConnnect);

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

query = "SELECT V.Код, U.ФИО, F.[Название], V.[Дата], V.[Состояние возврата] " +

"FROM ([Выдача Фильмов] as V INNER JOIN [Users] as U ON U.Код = V.[ID взявшего]) " +

"INNER JOIN [Films] as F ON F.Код = V.[ID фильма]";

ds = new DataSet();

da = new OleDbDataAdapter(query, myConnnect);

da.Fill(ds);

dataGridView2.DataSource = ds.Tables[0];

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

}

}

MissingFilms

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.Data.OleDb;

namespace ргр

{

public partial class MissingFilms : Form

{

OleDbConnection myConnnect;

OleDbDataAdapter da, da1;

DataSet ds, ds1;

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

{

MessageBox.Show("Cписок подан на печать");

this.Close();

}

public MissingFilms()

{

InitializeComponent();

string connect = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:/personal/учеба/4 семестр/Визуалка/ргр/ргр/ргр/videoteka.mdb;";

myConnnect = new OleDbConnection(connect);

DataLoad();

}

private void DataLoad()

{

try

{

string sqlstring = "SELECT \* FROM [Films] WHERE Наличие = False";

ds = new DataSet();

da = new OleDbDataAdapter(sqlstring, myConnnect);

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

}

}

AddDebtor

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 ргр

{

public partial class AddDebtor : Form

{

public AddDebtor(int AddUserID, int AddFilmID)

{

InitializeComponent();

}

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

{

this.Close();

}

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

{

archive Main = this.Owner as archive;

Main.AddUserID = Convert.ToInt16(textBox1.Text);

Main.AddFilmID = Convert.ToInt16(textBox2.Text);

this.Close();

}

}

}

Letter

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 ргр

{

public partial class Letter : Form

{

public Letter(string Name, string Address)

{

InitializeComponent();

label1.Text = "Уважаемый "+ Name;

label2.Text = Name;

label3.Text = Address;

DateTime thisDay = DateTime.Now;

label11.Text = thisDay.ToString("d");

}

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

{

if(checkBox1.Checked = true)

{

MessageBox.Show("Письмо подано на печать");

this.Close();

}

}

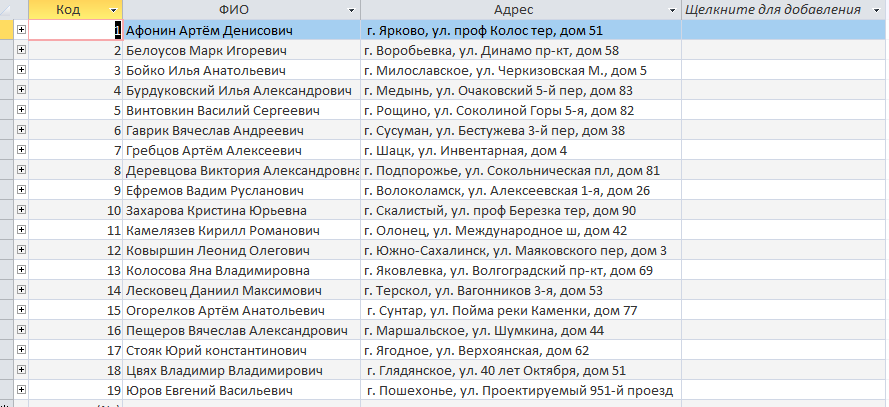
}

}

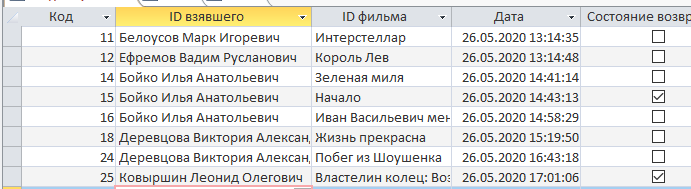
Работа программы



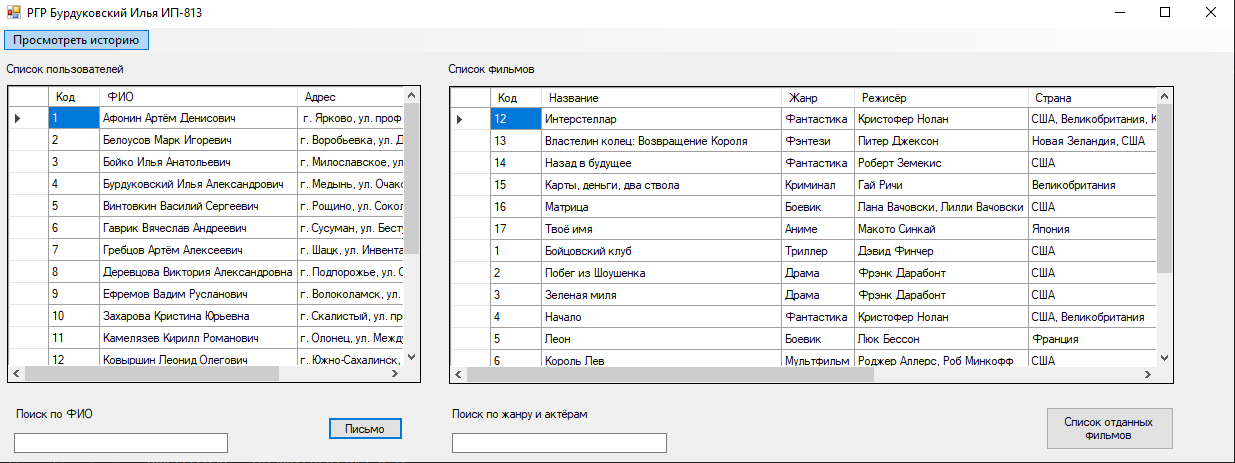
Созданная база данных (Данные о фильмах)



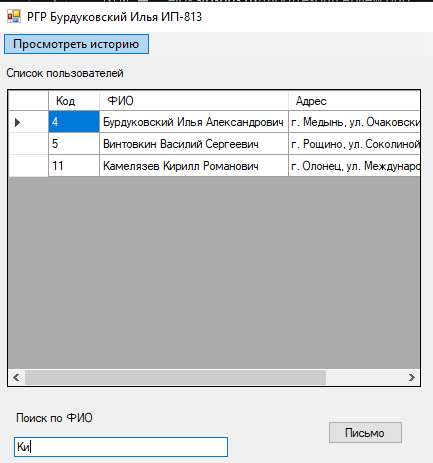
Созданная база данных (Данные о пользователях)



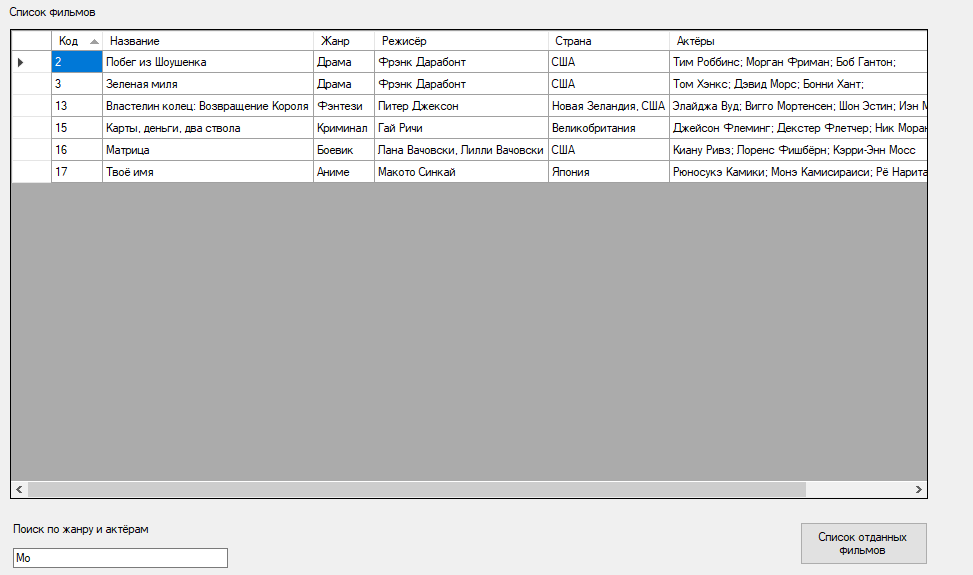
Созданная база данных (Архив выдач)



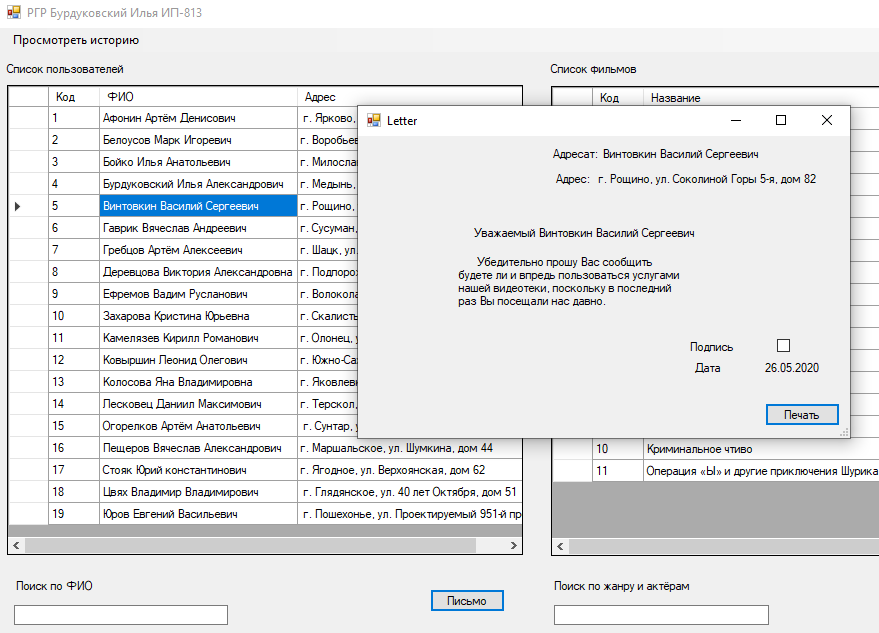
Первоначальная форма с данными из таблиц



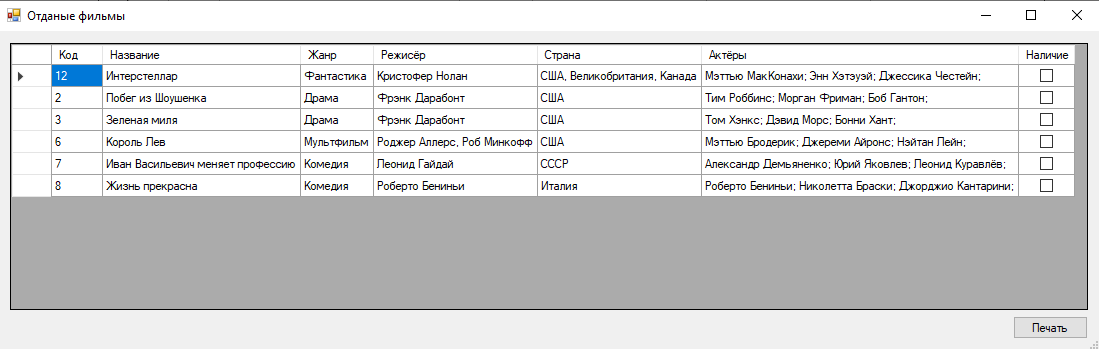
Неточный поиск по ФИО

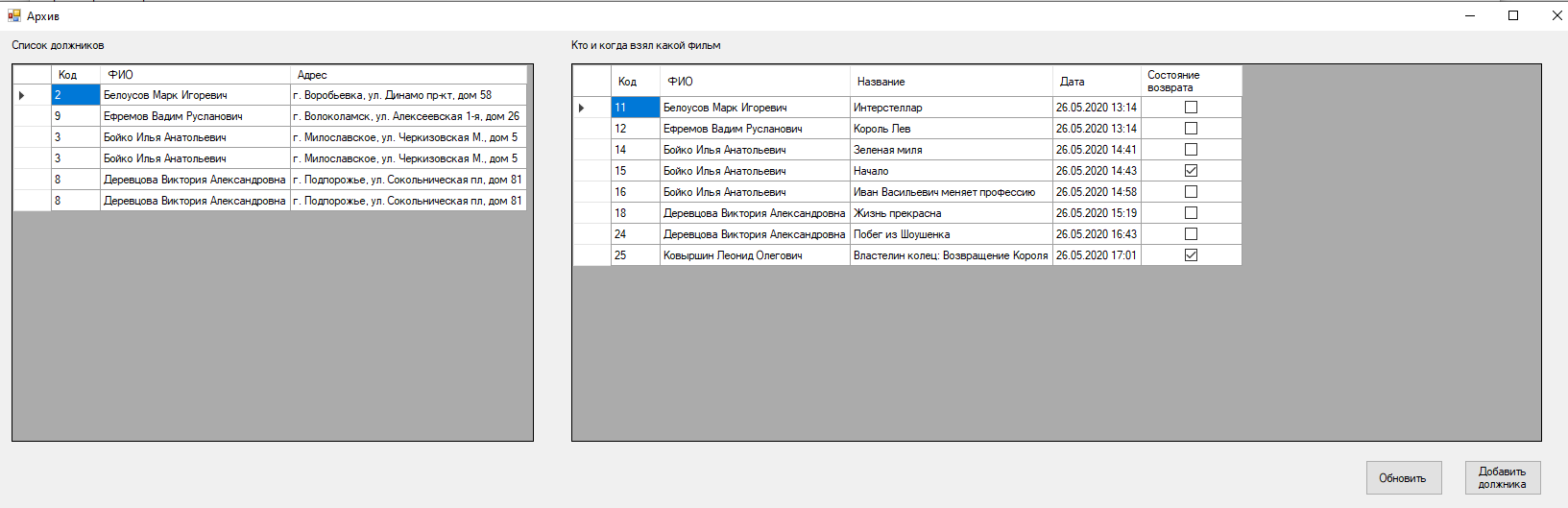


Поиск по жанрам и актёрам

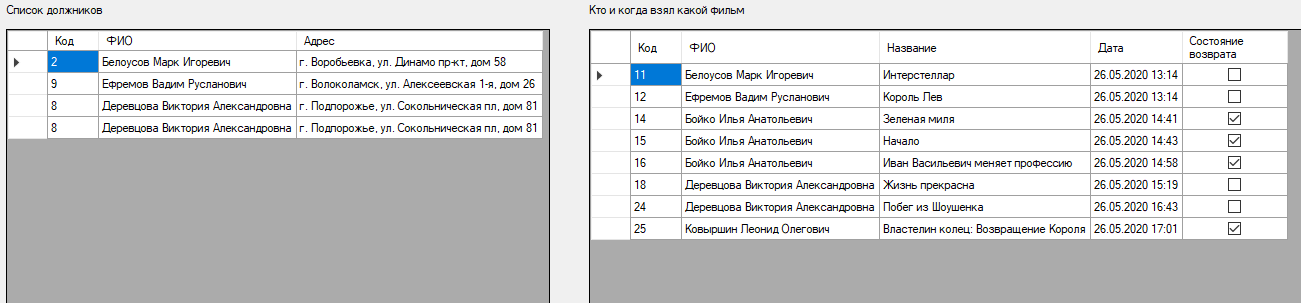


Письмо для выбранного пользователя

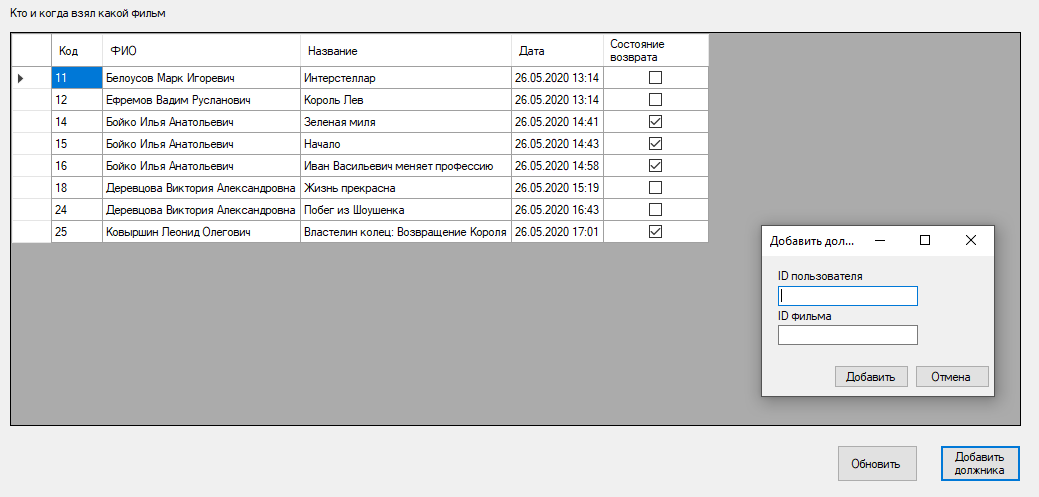


Список отданных фильмов

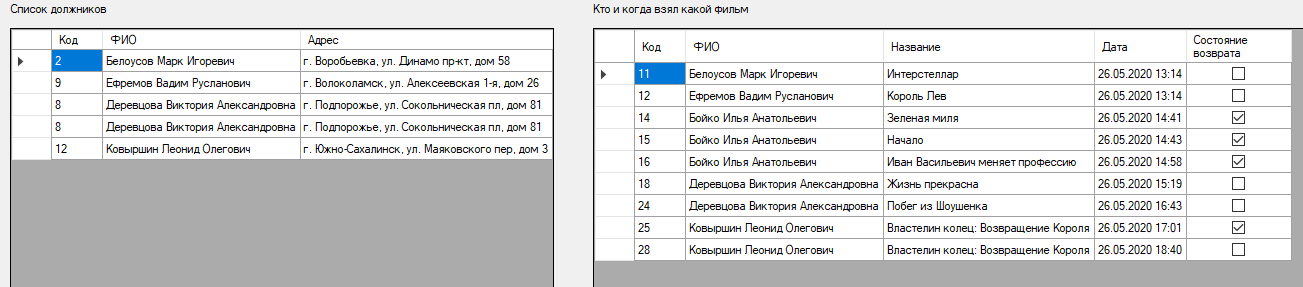
Архив

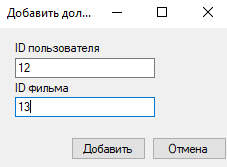


После возврата фильма

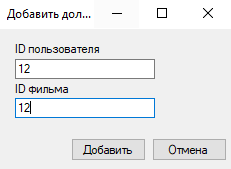


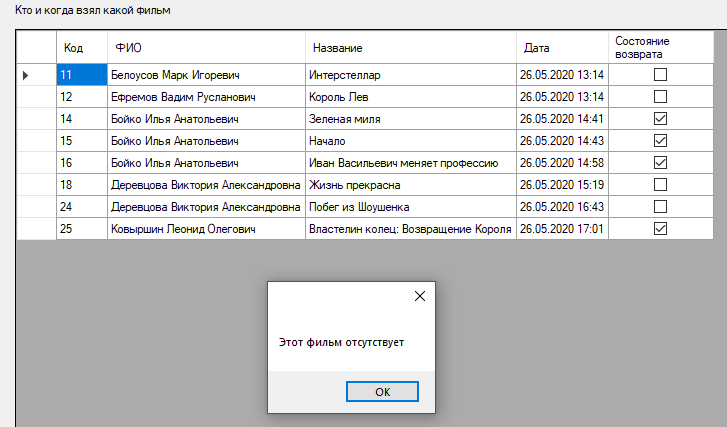
Добавление должника



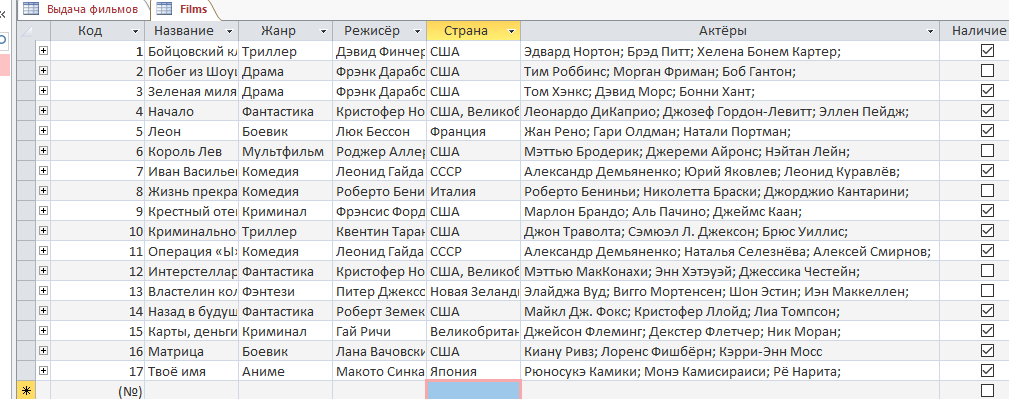


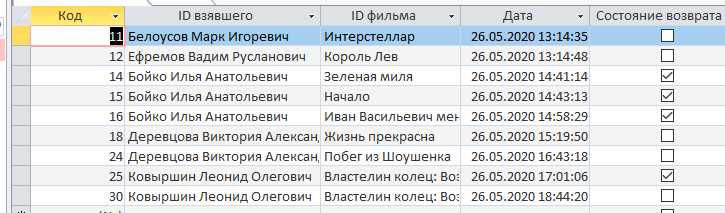
После добавления должника





В случае отсутствия





Обновлённые таблицы в БД