Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра прикладной математики и кибернетики

Расчетно-графическая работа

по дисциплине «Визуальное программирование»

«Видеотека»

Выполнил: студент 2 курса группы ИП-811 Мироненко К. А

Проверил: доцент кафедры ПМиК Ситняковская Е. И.

**Оглавление**

[1. Постановка задачи 3](#_Toc48826442)

[2. Примеры работы программы 5](#_Toc48826443)

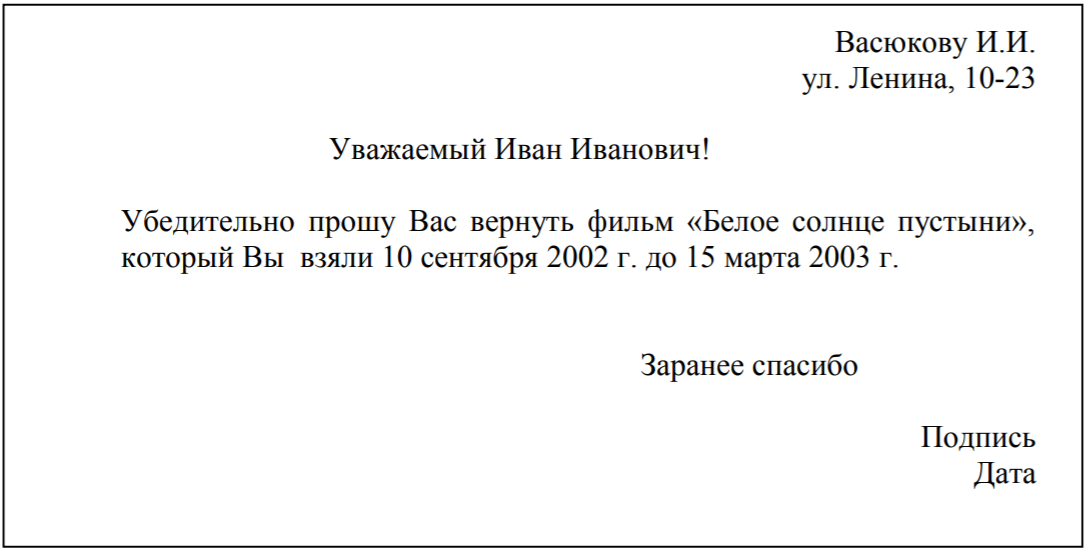
[*Приложение* Листинг 15](#_Toc48826444)

# Постановка задачи

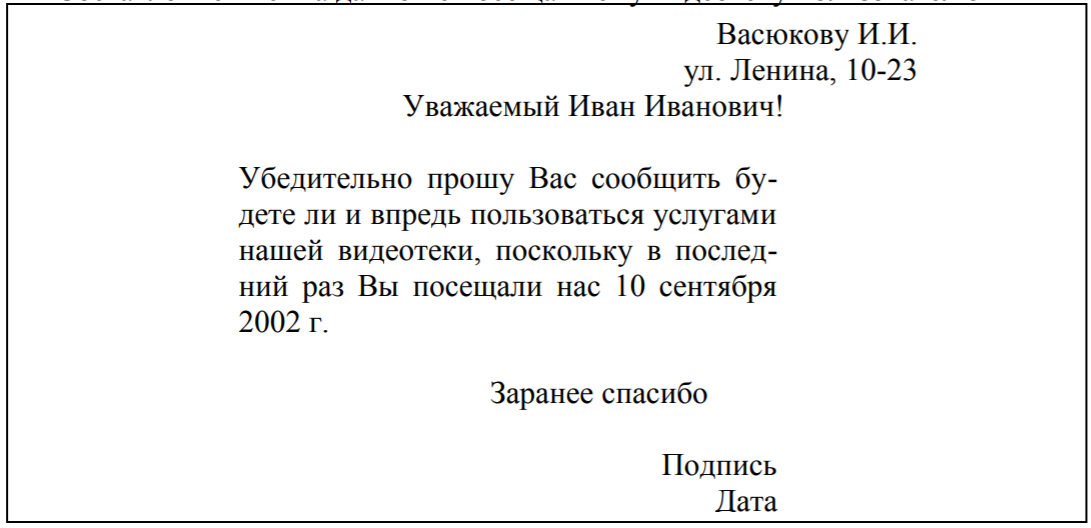
Некий любитель кино обладает огромным собранием фильмов (на видеокассетах, компакт-дисках, DVD-дисках), а также большим количеством друзей, которые тоже любят кино и берут фильмы, частенько забывая их вовремя отдавать. Поскольку и фильмов, и друзей слишком много, то наш любитель кино уже запутался, кто 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

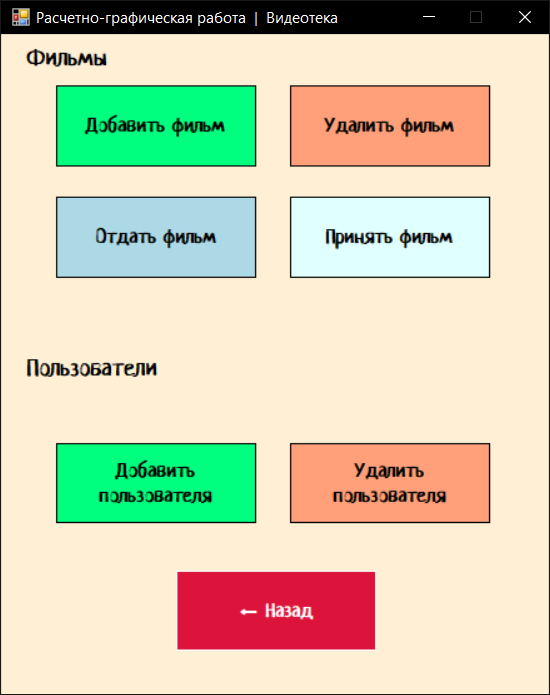


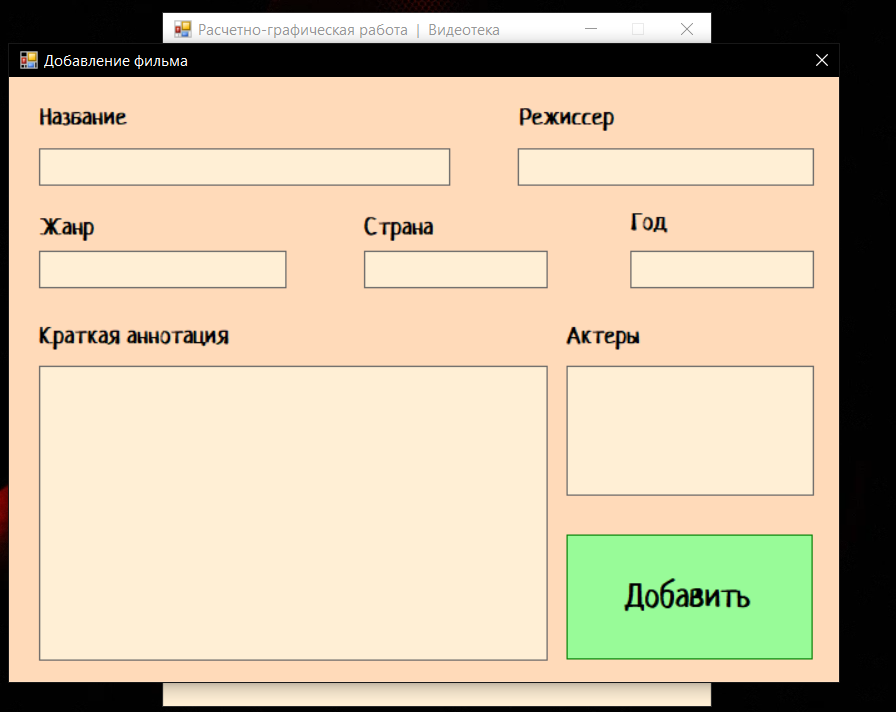
1. Составление письма давно не посещавшему видеотеку пользователю

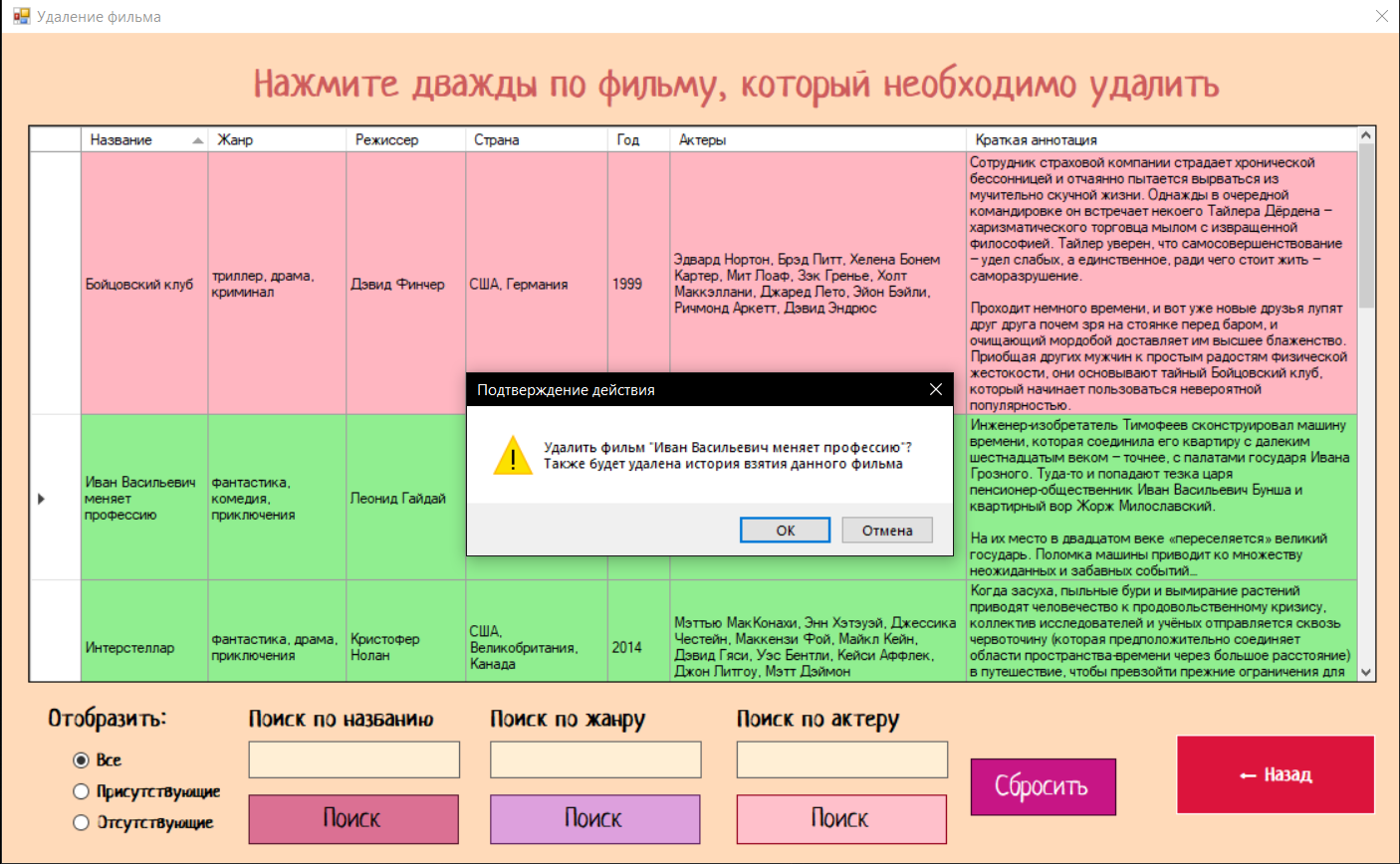


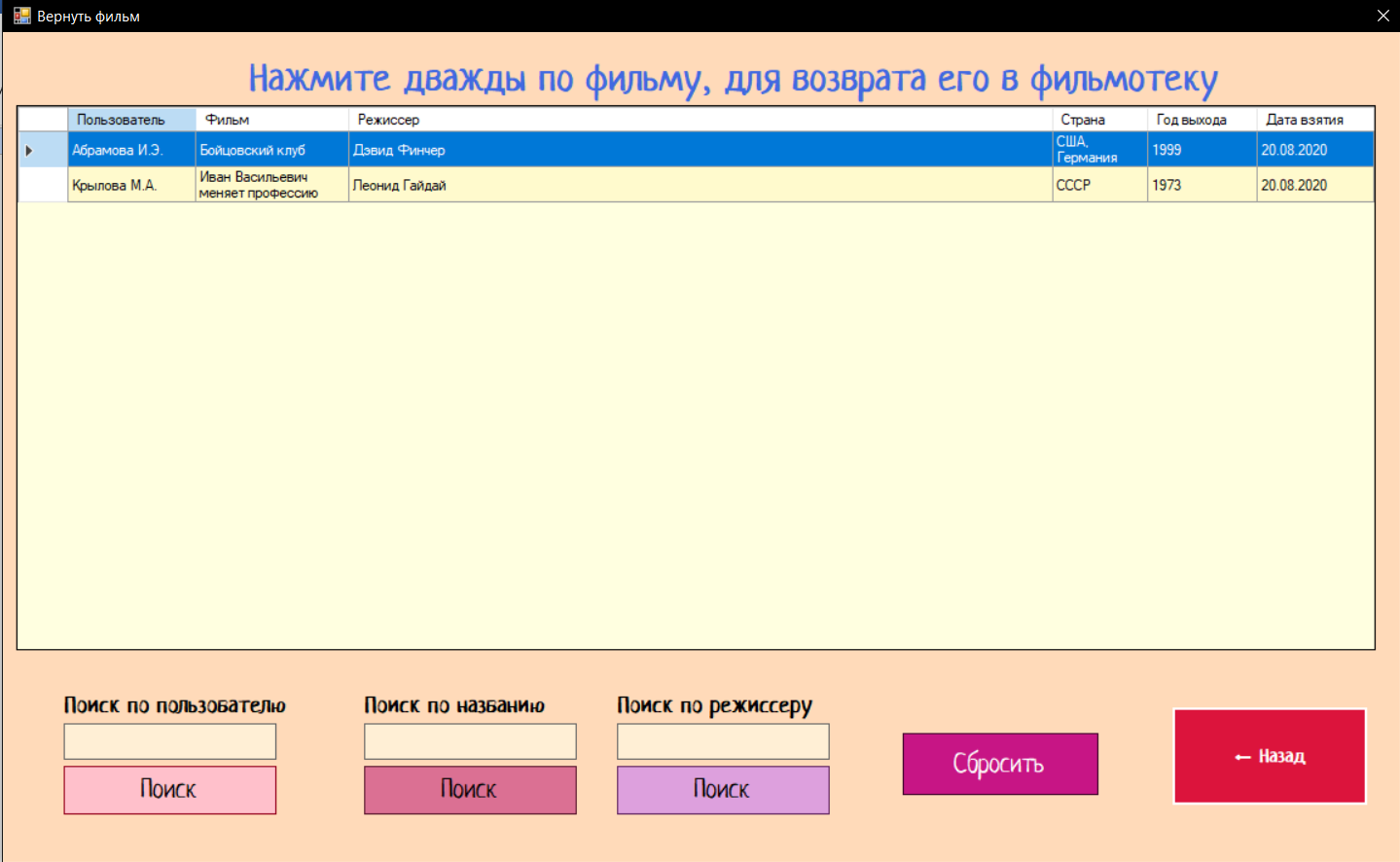
# Примеры работы программы

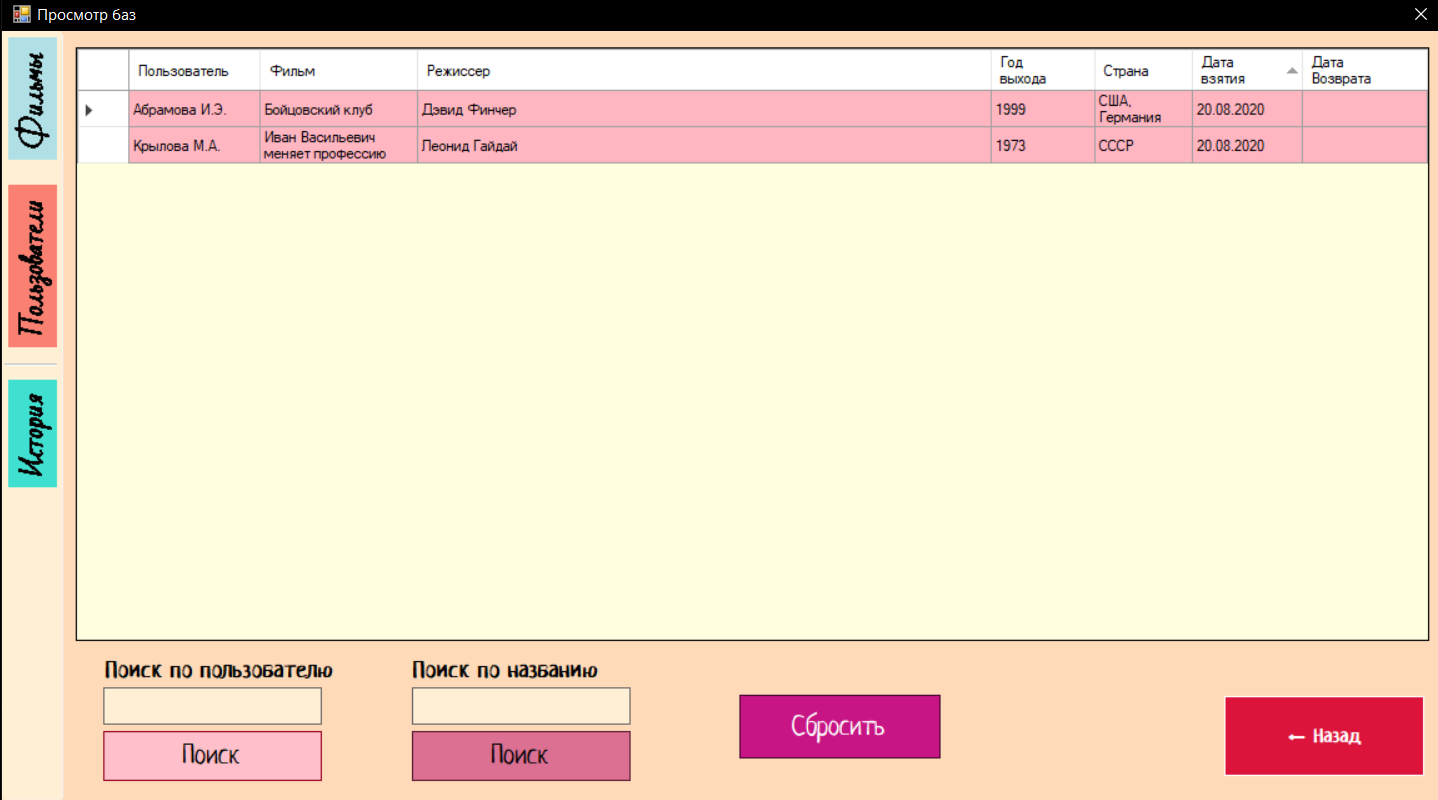


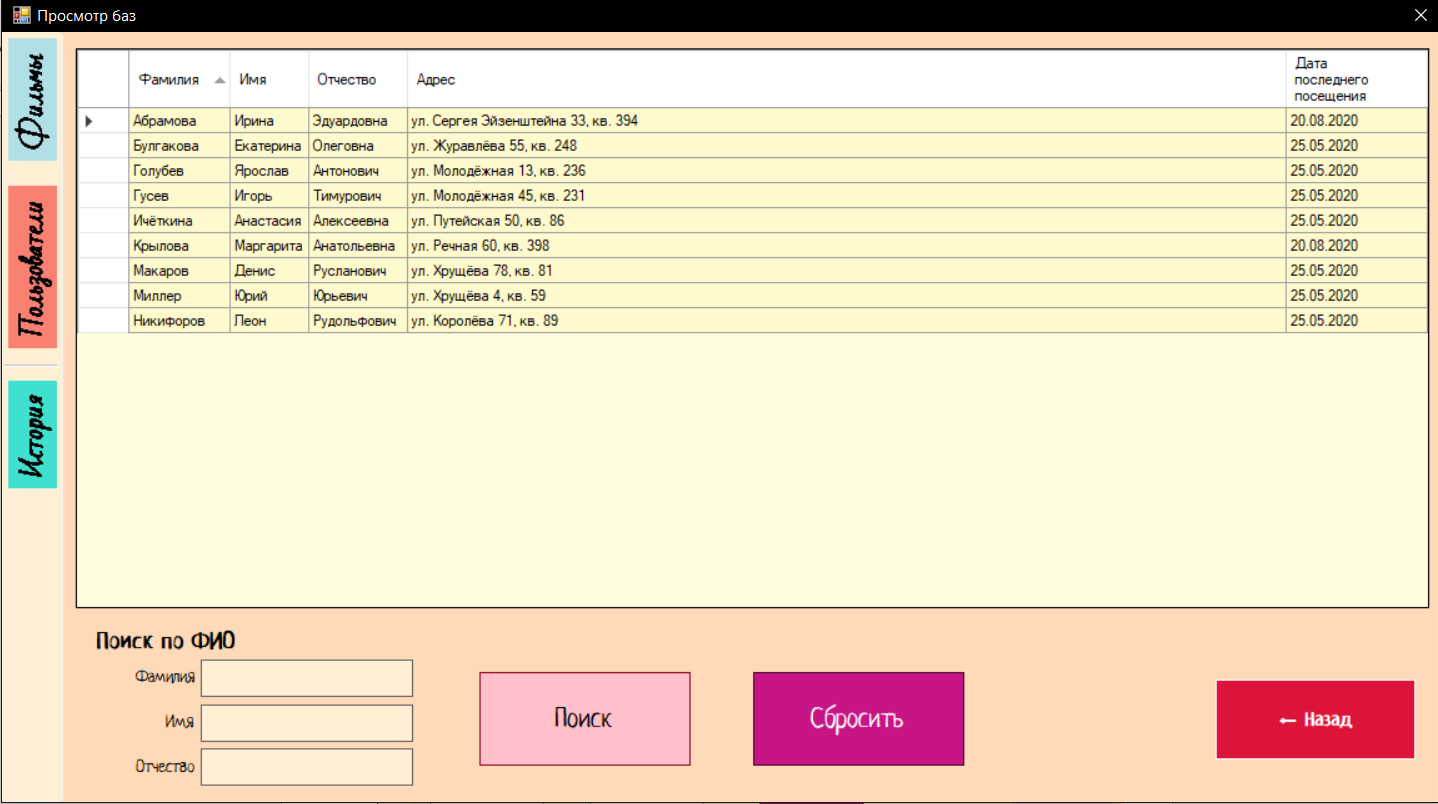


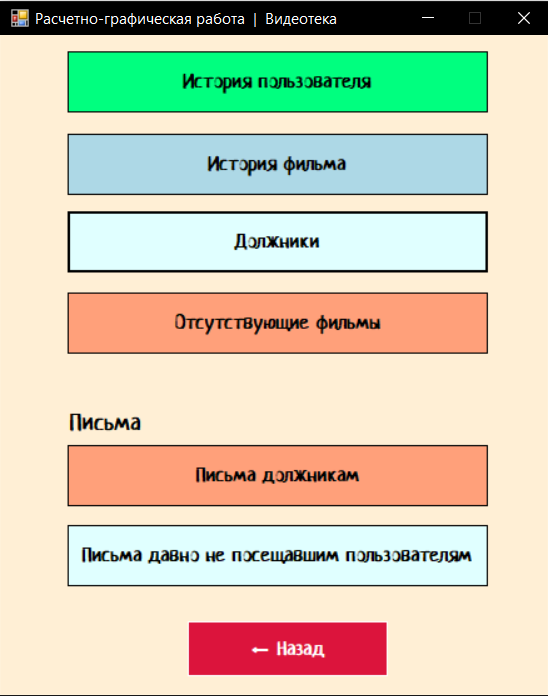


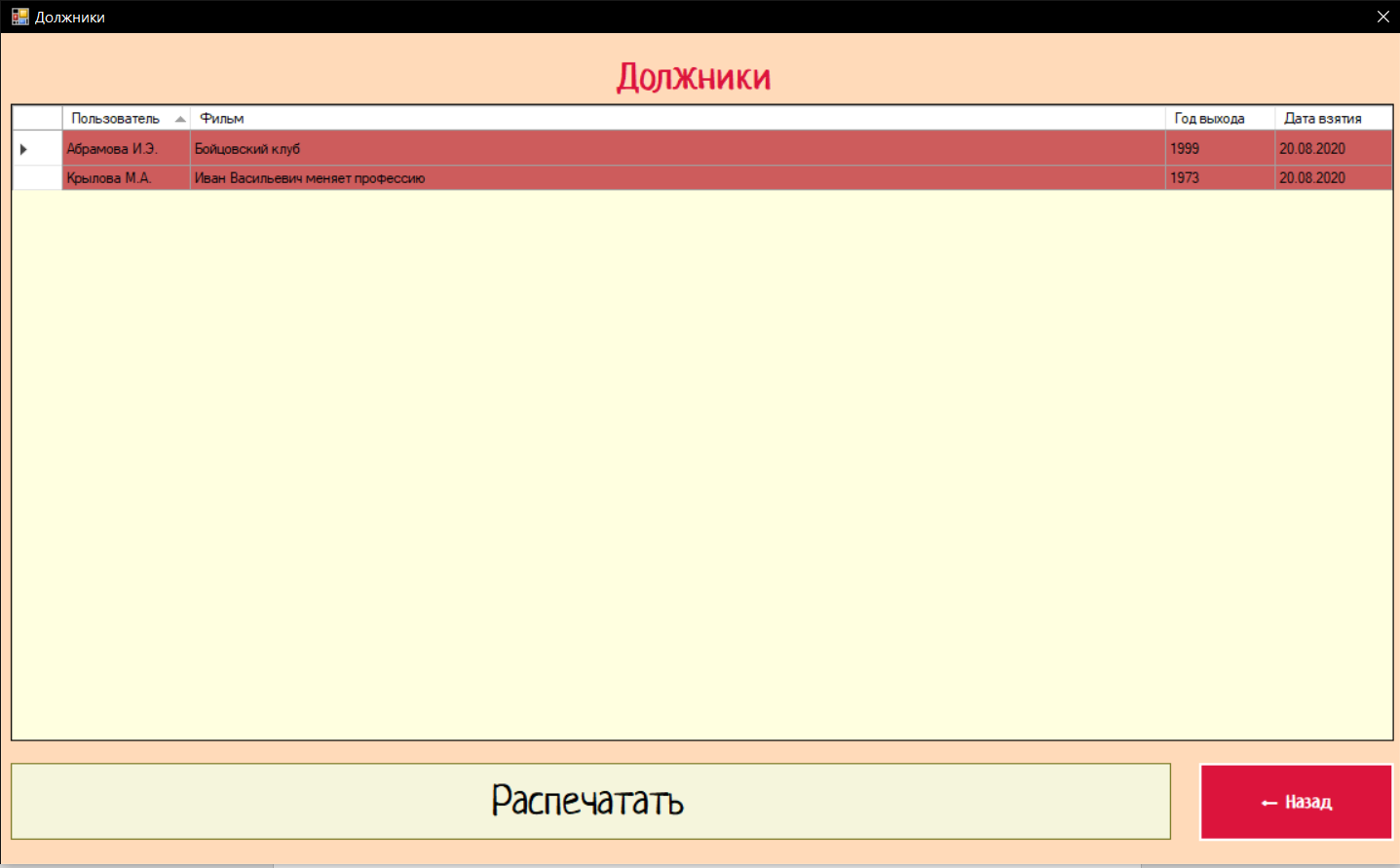


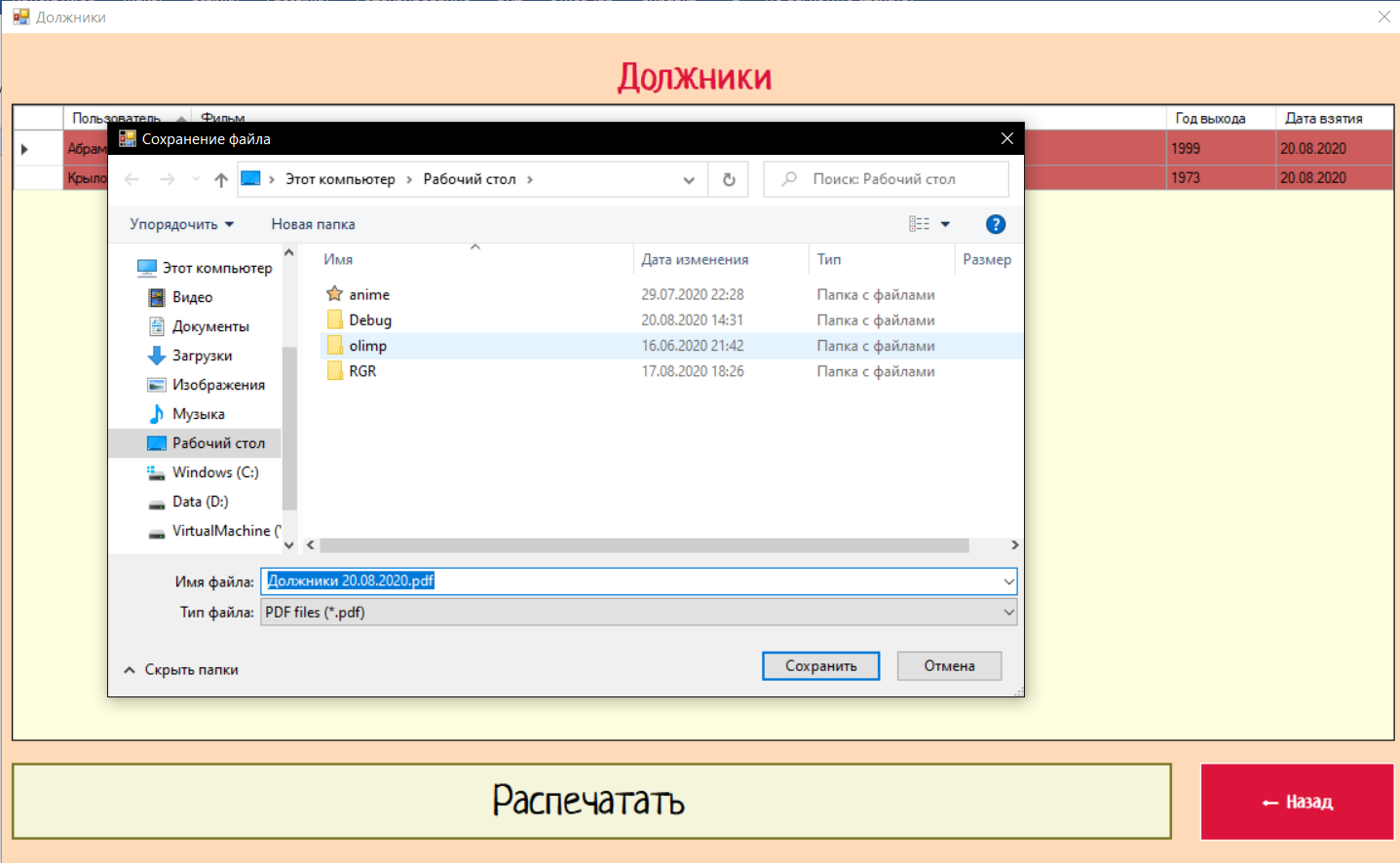


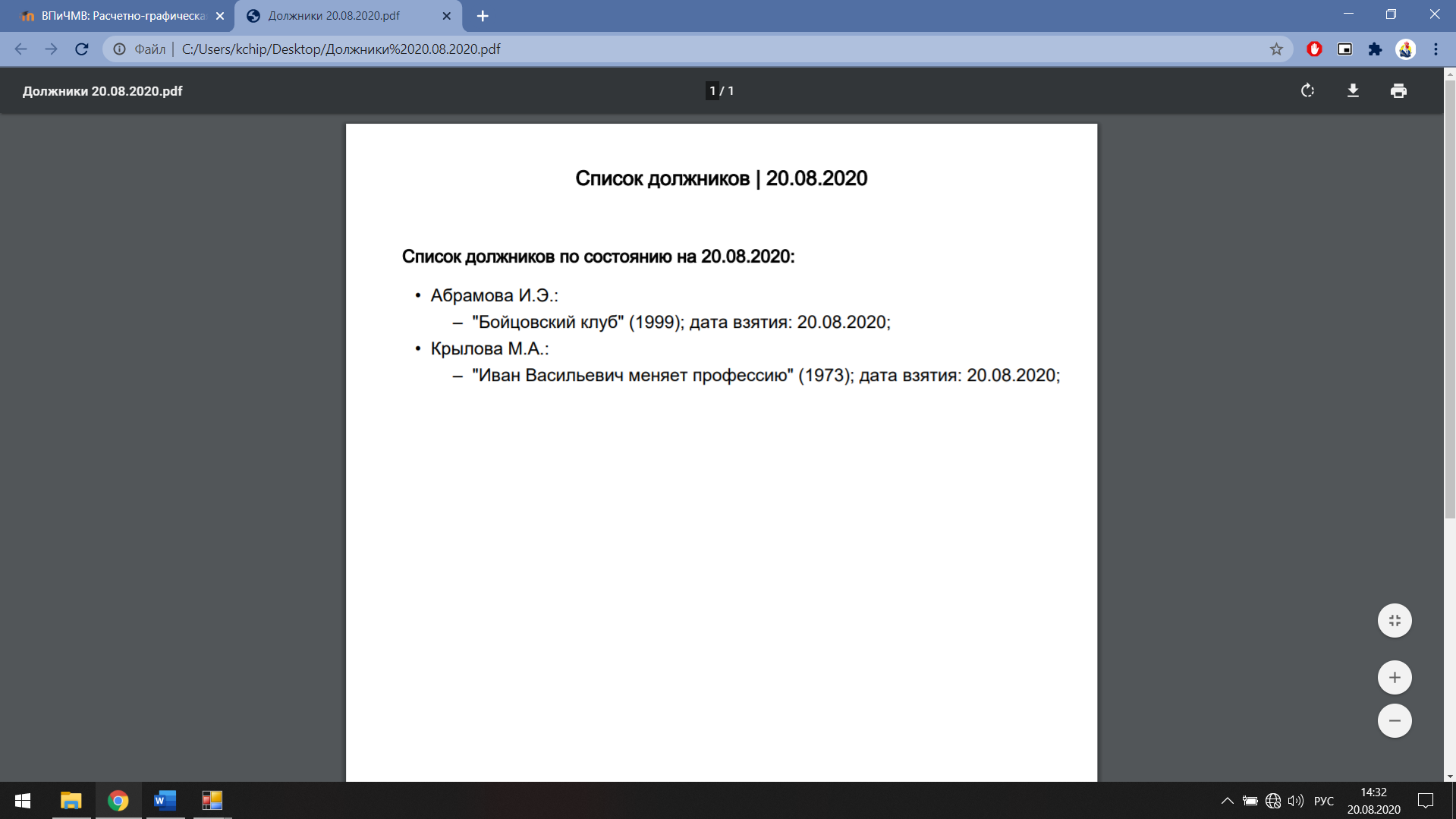


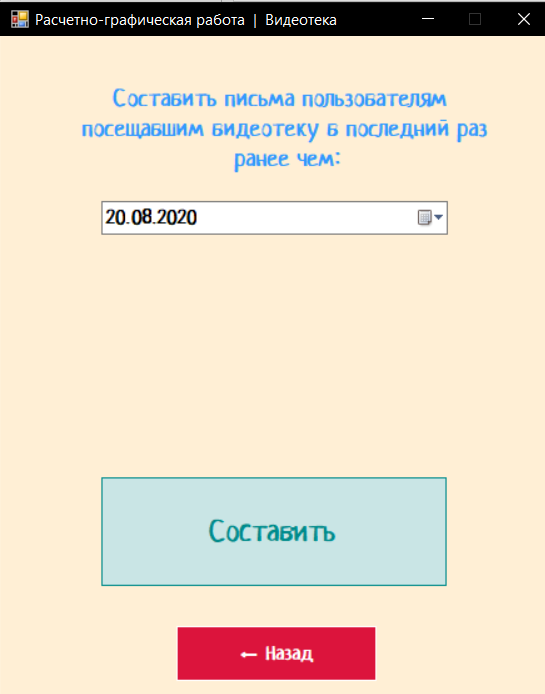


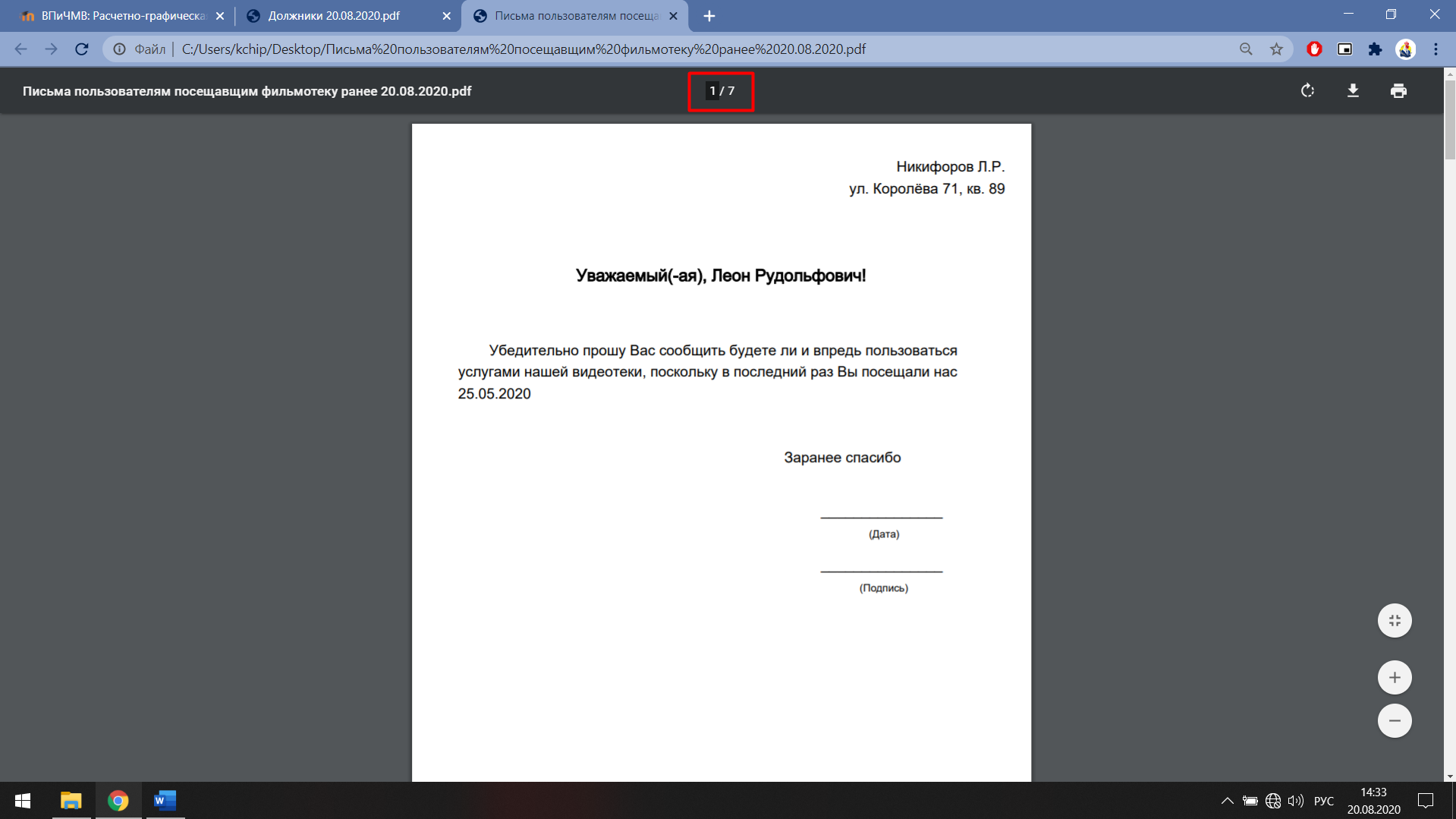












# *Приложение* Листинг

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace RGR

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new FormMain());

}

}

}

**FormMain.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using iTextSharp.text;

using iTextSharp.text.pdf;

namespace RGR

{

public partial class FormMain : Form

{

public FormMain()

{

InitializeComponent();

panelMain.Dock = System.Windows.Forms.DockStyle.Fill;

panelControl.Dock = System.Windows.Forms.DockStyle.Fill;

panelPrint.Dock = System.Windows.Forms.DockStyle.Fill;

panelLettersMissing.Dock = System.Windows.Forms.DockStyle.Fill;

panelLettersDebtors.Dock = System.Windows.Forms.DockStyle.Fill;

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Главное окно \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

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

{

panelMain.Hide();

panelControl.Show();

}

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

{

panelMain.Hide();

panelPrint.Show();

}

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

{

FormView form = new FormView

{

Owner = this

};

form.ShowDialog();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно управления \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

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

{

FormAddFilm form = new FormAddFilm

{

Owner = this

};

form.ShowDialog();

}

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

{

FormDelFilm form = new FormDelFilm

{

Owner = this

};

form.ShowDialog();

}

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

{

FormFilmGive form = new FormFilmGive

{

Owner = this

};

form.ShowDialog();

}

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

{

FormFilmReceive form = new FormFilmReceive

{

Owner = this

};

form.ShowDialog();

}

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

{

FormAddUser form = new FormAddUser

{

Owner = this

};

form.ShowDialog();

}

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

{

FormDelUser form = new FormDelUser

{

Owner = this

};

form.ShowDialog();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно печати \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

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

{

FormHistoryUser form = new FormHistoryUser

{

Owner = this

};

form.ShowDialog();

}

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

{

FormHistoryFilm form = new FormHistoryFilm

{

Owner = this

};

form.ShowDialog();

}

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

{

FormDebtors form = new FormDebtors

{

Owner = this

};

form.ShowDialog();

}

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

{

FormMissingFilms form = new FormMissingFilms

{

Owner = this

};

form.ShowDialog();

}

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

{

panelPrint.Hide();

panelLettersDebtors.Show();

dateTimePicker2.Format = DateTimePickerFormat.Short;

dateTimePicker2.MaxDate = DateTime.Now;

dateTimePicker2.Value = DateTime.Now;

dateTimePicker3.Format = DateTimePickerFormat.Short;

dateTimePicker3.MinDate = DateTime.Now;

dateTimePicker3.Value = DateTime.Now;

}

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

{

panelPrint.Hide();

panelLettersMissing.Show();

dateTimePicker1.Format = DateTimePickerFormat.Short;

dateTimePicker1.MaxDate = DateTime.Now;

dateTimePicker1.Value = DateTime.Now;

}

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

{

panelPrint.Hide();

panelControl.Hide();

panelMain.Show();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно давно не посещавших \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

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

{

пользователиTableAdapter.FillByLongVisited(this.databaseDataSet.Пользователи, dateTimePicker1.Value);

if (dataGridViewUser.Rows.Count == 0)

{

MessageBox.Show(

"Пользователей, посещавщих фильмотеку в последний раз ранее чем " + dateTimePicker1.Value.ToString().Split(' ')[0] + " не найдено!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

return;

}

SaveFileDialog dlg = new SaveFileDialog();

dlg.FileName = "Письма пользователям посещавщим фильмотеку ранее " + dateTimePicker1.Text;

dlg.Filter = "PDF files (\*.pdf)|\*.pdf";

dlg.RestoreDirectory = true;

dlg.Title = "Сохранение файла";

if (dlg.ShowDialog() == DialogResult.OK)

{

try

{

string outputFile = dlg.FileName;

FileStream fs = new FileStream(outputFile, FileMode.Create, FileAccess.Write, FileShare.None);

Document doc = new Document(PageSize.A4, 25, 25, 25, 15);

PdfWriter writer = PdfWriter.GetInstance(doc, fs);

BaseFont baseFont = BaseFont.CreateFont(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "ARIAL.TTF"), BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);

iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.NORMAL);

doc.Open();

foreach (DataGridViewRow row in dataGridViewUser.Rows)

{

string userF = row.Cells["фамилияDataGridViewTextBoxColumn"].Value.ToString();

string userI = row.Cells["имяDataGridViewTextBoxColumn"].Value.ToString();

string userO = row.Cells["отчествоDataGridViewTextBoxColumn"].Value.ToString();

string userInitials = userF + " " + userI.Substring(0, 1) + "." + userO.Substring(0, 1) + ".";

string adress = row.Cells["адресDataGridViewTextBoxColumn"].Value.ToString();

string date = row.Cells["датаПоследнегоПосещенияDataGridViewTextBoxColumn"].Value.ToString().Split(' ')[0];

Paragraph p = new Paragraph(userInitials + "\n" + adress, font);

p.SpacingBefore = 20;

p.SpacingAfter = 40;

p.Alignment = 2;

doc.Add(p);

p = new Paragraph("Уважаемый(-ая), " + userI + " " + userO + "!", new iTextSharp.text.Font(baseFont, 16, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.SpacingAfter = 30;

p.Alignment = 1;

doc.Add(p);

string text = "Убедительно прошу Вас сообщить будете ли и впредь пользоваться услугами нашей видеотеки, поскольку в последний раз Вы посещали нас ";

p = new Paragraph(text + date, font);

p.SpacingBefore = 20;

p.SpacingAfter = 20;

p.IndentationLeft = 20;

p.IndentationRight = 20;

p.FirstLineIndent = 30;

p.Alignment = 0;

doc.Add(p);

p = new Paragraph("Заранее спасибо", font);

p.SpacingBefore = 20;

p.SpacingAfter = 20;

p.IndentationRight = 100;

p.Alignment = 2;

doc.Add(p);

p = new Paragraph("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n", font);

p.Add(new Chunk("(Дата) ", new iTextSharp.text.Font(baseFont, 10, iTextSharp.text.Font.NORMAL)));

p.SpacingBefore = 10;

p.IndentationRight = 60;

p.Alignment = 2;

doc.Add(p);

p = new Paragraph("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n", font);

p.Add(new Chunk("(Подпись) ", new iTextSharp.text.Font(baseFont, 10, iTextSharp.text.Font.NORMAL)));

p.SpacingBefore = 10;

p.IndentationRight = 60;

p.Alignment = 2;

doc.Add(p);

doc.NewPage();

}

doc.Close();

writer.Close();

MessageBox.Show(

"Письма успешно составлены! \n\n Файл сохранен по следующей директории: " + outputFile,

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

catch

{

MessageBox.Show(

"Ошибка! \n Не удалось сохранить файл! \n Проверьте права на запись в дирректории для сохранения и попробуйте снова!",

"Упс! Что-то пошло не так...",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

}

dlg.Dispose();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно должников \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

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

{

историяTableAdapter.FillByDateTake(this.databaseDataSet.История, dateTimePicker2.Value);

this.dataGridViewHistory.Sort(this.dataGridViewHistory.Columns[пользовательDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

if (dataGridViewHistory.Rows.Count == 0)

{

MessageBox.Show(

"Должников, взявших фильмы ранее чем " + dateTimePicker2.Value.ToString().Split(' ')[0] + " не найдено!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

return;

}

SaveFileDialog dlg = new SaveFileDialog();

dlg.FileName = "Письма должникам, которые брали фильмы ранее " + dateTimePicker2.Text;

dlg.Filter = "PDF files (\*.pdf)|\*.pdf";

dlg.RestoreDirectory = true;

dlg.Title = "Сохранение файла";

if (dlg.ShowDialog() == DialogResult.OK)

{

try

{

string outputFile = dlg.FileName;

FileStream fs = new FileStream(outputFile, FileMode.Create, FileAccess.Write, FileShare.None);

Document doc = new Document(PageSize.A4, 25, 25, 25, 15);

PdfWriter writer = PdfWriter.GetInstance(doc, fs);

BaseFont baseFont = BaseFont.CreateFont(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "ARIAL.TTF"), BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);

iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.NORMAL);

doc.Open();

DataGridView dgv = dataGridViewHistory;

while(dgv.Rows.Count != 0)

{

Int32 userID = Int32.Parse(dgv.Rows[0].Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString());

string userInitials = dgv.Rows[0].Cells["пользовательDataGridViewTextBoxColumn"].Value.ToString();

string date = dateTimePicker3.Text;

пользователиTableAdapter.FillByUserCode(this.databaseDataSet.Пользователи, userID);

// Говнокод во всей красе

string userI = dataGridViewUser.Rows[0].Cells["имяDataGridViewTextBoxColumn"].Value.ToString();

string userO = dataGridViewUser.Rows[0].Cells["отчествоDataGridViewTextBoxColumn"].Value.ToString();

string adress = dataGridViewUser.Rows[0].Cells["адресDataGridViewTextBoxColumn"].Value.ToString();

Paragraph p = new Paragraph(userInitials + "\n" + adress, font);

p.SpacingBefore = 20;

p.SpacingAfter = 40;

p.Alignment = 2;

doc.Add(p);

p = new Paragraph("Уважаемый(-ая), " + userI + " " + userO + "!", new iTextSharp.text.Font(baseFont, 16, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.SpacingAfter = 10;

p.Alignment = 1;

doc.Add(p);

string text = "Убедительно прошу Вас вернуть следующие фильмы до ";

p = new Paragraph(text + date + "г. :\n", font);

p.SpacingBefore = 20;

p.SpacingAfter = 10;

p.IndentationLeft = 20;

p.IndentationRight = 20;

p.FirstLineIndent = 30;

p.Alignment = 0;

doc.Add(p);

string tempCode = dgv.Rows[0].Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString();

iTextSharp.text.List list = new iTextSharp.text.List(List.UNORDERED);

list.IndentationLeft = 80;

list.SetListSymbol("\u2022");

while ((dgv.Rows.Count != 0) && (tempCode == dgv.Rows[0].Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString()))

{

DataGridViewRow row = dgv.Rows[0];

string filmName = row.Cells["фильмDataGridViewTextBoxColumn"].Value.ToString();

string filmYear = row.Cells["годВыходаDataGridViewTextBoxColumn"].Value.ToString();

string dateTake = row.Cells["датаВзятияDataGridViewTextBoxColumn"].Value.ToString().Split(' ')[0];

list.Add(new iTextSharp.text.ListItem(" \"" + filmName + "\" (" + filmYear + "), который Вы взяли " + dateTake + ";", font));

dgv.Rows.RemoveAt(0);

}

doc.Add(list);

p = new Paragraph("Заранее спасибо", font);

p.SpacingBefore = 40;

p.SpacingAfter = 40;

p.IndentationRight = 120;

p.Alignment = 2;

doc.Add(p);

p = new Paragraph("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n", font);

p.Add(new Chunk("(Дата) ", new iTextSharp.text.Font(baseFont, 10, iTextSharp.text.Font.NORMAL)));

p.SpacingBefore = 10;

p.IndentationRight = 60;

p.Alignment = 2;

doc.Add(p);

p = new Paragraph("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n", font);

p.Add(new Chunk("(Подпись) ", new iTextSharp.text.Font(baseFont, 10, iTextSharp.text.Font.NORMAL)));

p.SpacingBefore = 10;

p.IndentationRight = 60;

p.Alignment = 2;

doc.Add(p);

doc.NewPage();

}

doc.Close();

writer.Close();

MessageBox.Show(

"Письма успешно составлены! \n\n Файл сохранен по следующей директории: " + outputFile,

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

catch

{

MessageBox.Show(

"Ошибка! \n Не удалось сохранить файл! \n Проверьте права на запись в дирректории для сохранения и попробуйте снова!",

"Упс! Что-то пошло не так...",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

}

dlg.Dispose();

}

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

{

panelPrint.Show();

panelLettersMissing.Hide();

panelLettersDebtors.Hide();

}

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

{

this.историяTableAdapter.Fill(this.databaseDataSet.История);

}

}

}

**FormAddFilm.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 RGR

{

public partial class FormAddFilm : Form

{

public FormAddFilm()

{

InitializeComponent();

}

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

{

if (textBoxName.Text != "" && textBoxGenre.Text != "" && textBoxYear.Text != "" && textBoxActors.Text != ""

&& textBoxAnnotation.Text != "" && textBoxCountry.Text != "" && textBoxProducer.Text != "")

{

if (textBoxYear.TextLength != 4)

{

MessageBox.Show(

"Некорректный год выхода фильма",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

else

{

try

{

int tmp = Convert.ToInt32(textBoxYear.Text);

if (tmp < 1888)

{

throw new Exception();

}

try

{

фильмыTableAdapter.AddFilm(textBoxName.Text, textBoxGenre.Text, textBoxProducer.Text, textBoxCountry.Text, textBoxYear.Text, textBoxActors.Text, textBoxAnnotation.Text, false);

MessageBox.Show(

"Запись успешно добавлена!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

Close();

}

catch (Exception)

{

MessageBox.Show(

"Произошла ошибка, попробуйте снова",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

}

catch (Exception)

{

MessageBox.Show(

"Некорректный год выхода фильма!",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

}

}

else

{

MessageBox.Show(

"Некорректные данные, все поля должны быть заполнены!",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

}

}

}

**FormAddUser.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 RGR

{

public partial class FormAddFilm : Form

{

public FormAddFilm()

{

InitializeComponent();

}

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

{

if (textBoxName.Text != "" && textBoxGenre.Text != "" && textBoxYear.Text != "" && textBoxActors.Text != ""

&& textBoxAnnotation.Text != "" && textBoxCountry.Text != "" && textBoxProducer.Text != "")

{

if (textBoxYear.TextLength != 4)

{

MessageBox.Show(

"Некорректный год выхода фильма",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

else

{

try

{

int tmp = Convert.ToInt32(textBoxYear.Text);

if (tmp < 1888)

{

throw new Exception();

}

try

{

фильмыTableAdapter.AddFilm(textBoxName.Text, textBoxGenre.Text, textBoxProducer.Text, textBoxCountry.Text, textBoxYear.Text, textBoxActors.Text, textBoxAnnotation.Text, false);

MessageBox.Show(

"Запись успешно добавлена!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

Close();

}

catch (Exception)

{

MessageBox.Show(

"Произошла ошибка, попробуйте снова",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

}

catch (Exception)

{

MessageBox.Show(

"Некорректный год выхода фильма!",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

}

}

else

{

MessageBox.Show(

"Некорректные данные, все поля должны быть заполнены!",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

}

}

}

**FormDelFilm.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 RGR

{

public partial class FormDelFilm : Form

{

public FormDelFilm()

{

InitializeComponent();

}

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

{

this.фильмыTableAdapter.Fill(this.databaseDataSet.Фильмы);

this.dataGridViewFilm.Sort(this.dataGridViewFilm.Columns[названиеDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно фильмов \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

private void RadioBtnFilm\_CheckedChanged(object sender, EventArgs e)

{

if (radioBtnFilmAll.Checked)

фильмыTableAdapter.Fill(this.databaseDataSet.Фильмы);

else if (radioBtnFilmPresent.Checked)

фильмыTableAdapter.FillByAbsent(this.databaseDataSet.Фильмы, false);

else if (radioBtnFilmAbsent.Checked)

фильмыTableAdapter.FillByAbsent(this.databaseDataSet.Фильмы, true);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

// Выделение цветом

private void DataGridViewFilm\_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)

{

foreach (DataGridViewRow row in dataGridViewFilm.Rows)

{

if ((bool)row.Cells["отсутствуетDataGridViewCheckBoxColumn"].Value)

dataGridViewFilm.Rows[row.Index].DefaultCellStyle.BackColor = Color.LightPink;

else

dataGridViewFilm.Rows[row.Index].DefaultCellStyle.BackColor = Color.LightGreen;

}

}

// Поиск фильма

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

{

string btnName = (sender as Button).Name.ToString();

if (radioBtnFilmPresent.Checked)

{

var condition = MessageBox.Show(

"Поиск будет выполнен только по присутствующим фильмам",

"Предупреждение",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName2(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text, (bool)false);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre2(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text, (bool)false);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor2(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text, (bool)false);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

break;

}

}

else if (radioBtnFilmAbsent.Checked)

{

var condition = MessageBox.Show(

"Поиск будет выполнен только по отсутствующим фильмам",

"Предупреждение",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName2(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text, (bool)true);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre2(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text, (bool)true);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor2(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text, (bool)true);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

break;

}

}

else

{

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

}

}

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

{

Close();

}

private void DataGridViewFilm\_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)

{

Int32 codeFilm = (int)dataGridViewFilm["кодФильма", dataGridViewFilm.CurrentRow.Index].Value;

string nameFilm = dataGridViewFilm["названиеDataGridViewTextBoxColumn", dataGridViewFilm.CurrentRow.Index].Value.ToString();

DialogResult condition;

if ((bool)dataGridViewFilm["отсутствуетDataGridViewCheckBoxColumn", dataGridViewFilm.CurrentRow.Index].Value)

{

историяTableAdapter.FillByAbsentFilmCode(this.databaseDataSet.История, codeFilm);

string userName = dataGridViewDel["Пользователь", 0].Value.ToString();

condition = MessageBox.Show(

"Удалить отсутствующий фильм \"" + nameFilm + "\"? Фильм будет удален с рук пользователя \""+userName+"\", а также будет удалена история взятия данного фильма",

"Подтверждение действия",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

}

else

{

condition = MessageBox.Show(

"Удалить фильм \""+ nameFilm + "\"? Также будет удалена история взятия данного фильма",

"Подтверждение действия",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

}

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

фильмыTableAdapter.DelFilm(codeFilm);

историяTableAdapter.DelRecordWhereFilm(codeFilm);

MessageBox.Show(

"Действие успешно выполнено!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

this.фильмыTableAdapter.Fill(this.databaseDataSet.Фильмы);

break;

}

}

private void dataGridViewFilm\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewFilm.ClearSelection();

}

}

}

**FormDelUser.cs**

using RGR.DatabaseDataSetTableAdapters;

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 RGR

{

public partial class FormDelUser : Form

{

public FormDelUser()

{

InitializeComponent();

}

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

{

this.историяTableAdapter.Fill(this.databaseDataSet.История);

this.пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

this.dataGridViewUser.Sort(this.dataGridViewUser.Columns[фамилияDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно пользователей \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

// Поиск пользователя

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

{

пользователиTableAdapter.FillByFIO(this.databaseDataSet.Пользователи, this.textBoxSearchUserF.Text, this.textBoxSearchUserI.Text, this.textBoxSearchUserO.Text);

}

// Сброс поиска

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

{

пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

this.textBoxSearchUserF.Text = null;

this.textBoxSearchUserI.Text = null;

this.textBoxSearchUserO.Text = null;

}

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

{

Close();

}

private void DataGridViewUser\_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)

{

Int32 userCode = (int)dataGridViewUser["кодПользователя", dataGridViewUser.CurrentRow.Index].Value;

MessageBox.Show(userCode.ToString());

string user = dataGridViewUser.CurrentRow.Cells["фамилияDataGridViewTextBoxColumn"].Value.ToString() + " " + dataGridViewUser.CurrentRow.Cells["имяDataGridViewTextBoxColumn"].Value.ToString().Substring(0, 1) + "." + dataGridViewUser.CurrentRow.Cells["отчествоDataGridViewTextBoxColumn"].Value.ToString().Substring(0, 1) + ".";

DialogResult condition;

историяTableAdapter.FillByAbsentUserCode(this.databaseDataSet.История, userCode);

if (dataGridViewDel.RowCount == 0)

{

condition = MessageBox.Show(

"Удалить пользователя \"" + user + "\"? Также будет удалена история пользования фильмотекой",

"Подтверждение действия",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

}

else

{

string films = "";

foreach (DataGridViewRow row in dataGridViewDel.Rows)

{

films += "\"" + row.Cells["Фильм"].Value.ToString() + "\"; ";

}

condition = MessageBox.Show(

"Удалить пользователя \"" + user + "\"? На руках у пользователя следующе фильмы: " + films + ", они будут удалены. Также будет очищена история взятия данного фильма",

"Подтверждение действия",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

}

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

пользователиTableAdapter.DelUser(userCode);

историяTableAdapter.DelRecordWhereUser(userCode);

foreach (DataGridViewRow row in dataGridViewDel.Rows)

{

фильмыTableAdapter.DelFilm((int)row.Cells["КодФильма"].Value);

}

MessageBox.Show(

"Действие успешно выполнено!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

this.пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

break;

}

}

private void dataGridViewUser\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewUser.ClearSelection();

}

}

}

**FormFilmGive.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 RGR

{

public partial class FormFilmGive : Form

{

public FormFilmGive()

{

InitializeComponent();

}

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

{

this.пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

this.фильмыTableAdapter.FillByAbsent(this.databaseDataSet.Фильмы, false);

this.историяTableAdapter.Fill(this.databaseDataSet.История);

this.dataGridViewFilm.Sort(this.dataGridViewFilm.Columns[названиеDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

this.dataGridViewUser.Sort(this.dataGridViewUser.Columns[фамилияDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

this.panelFilm.Dock = System.Windows.Forms.DockStyle.Fill;

this.panelUser.Dock = System.Windows.Forms.DockStyle.Fill;

panelUser.Hide();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно фильмов \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

// Поиск фильма

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

{

string btnName = (sender as Button).Name.ToString();

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName2(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text, (bool)false);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre2(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text, (bool)false);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor2(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text, (bool)false);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

}

// Убирает выделение строк

private void DataGridViewFilm\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewFilm.ClearSelection();

}

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

{

Close();

}

private void DataGridViewFilm\_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)

{

panelFilm.Hide();

panelUser.Show();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно пользователей \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

// Поиск пользователя

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

{

пользователиTableAdapter.FillByFIO(this.databaseDataSet.Пользователи, this.textBoxSearchUserF.Text, this.textBoxSearchUserI.Text, this.textBoxSearchUserO.Text);

}

// Сброс поиска

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

{

пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

this.textBoxSearchUserF.Text = null;

this.textBoxSearchUserI.Text = null;

this.textBoxSearchUserO.Text = null;

}

// Убирает выделение строк

private void DataGridViewUser\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewUser.ClearSelection();

}

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

{

panelFilm.Show();

panelUser.Hide();

}

private void DataGridViewUser\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

int codeUser = Convert.ToInt32(dataGridViewUser.CurrentRow.Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString());

int codeFilm = Convert.ToInt32(dataGridViewFilm.CurrentRow.Cells["кодФильмаDataGridViewTextBoxColumn"].Value.ToString());

string user = dataGridViewUser.CurrentRow.Cells["фамилияDataGridViewTextBoxColumn"].Value.ToString() + " " + dataGridViewUser.CurrentRow.Cells["имяDataGridViewTextBoxColumn"].Value.ToString().Substring(0, 1) + "." + dataGridViewUser.CurrentRow.Cells["отчествоDataGridViewTextBoxColumn"].Value.ToString().Substring(0, 1) + ".";

string filmName = dataGridViewFilm.CurrentRow.Cells["названиеDataGridViewTextBoxColumn"].Value.ToString();

string filmCountry = dataGridViewFilm.CurrentRow.Cells["странаDataGridViewTextBoxColumn"].Value.ToString();

string filmProducer = dataGridViewFilm.CurrentRow.Cells["режиссерDataGridViewTextBoxColumn"].Value.ToString();

string filmYear = dataGridViewFilm.CurrentRow.Cells["годDataGridViewTextBoxColumn"].Value.ToString();

var condition = MessageBox.Show(

"Отдать фильм: \n Название: \"" + filmName + "\"\n Режиссер(-ы): " + filmProducer + "\n Год выхода: " + filmYear + "\n Пользователю: " + user,

"Подтверждение действия",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

try

{

фильмыTableAdapter.UpdateAbsence(true, codeFilm);

пользователиTableAdapter.UpdateLastVisit(DateTime.Now.Date, codeUser);

историяTableAdapter.AddRecord(codeUser, codeFilm, user, filmName, filmProducer, filmCountry, filmYear, DateTime.Now.Date, null);

MessageBox.Show(

"Действие успешно выполнено!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

фильмыTableAdapter.FillByAbsent(this.databaseDataSet.Фильмы, false);

panelFilm.Show();

panelUser.Hide();

}

catch (Exception)

{

MessageBox.Show(

"Произошла ошибка, попробуйте снова",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

break;

}

}

}

}

**FormFilmReceive.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 RGR

{

public partial class FormFilmReceive : Form

{

public FormFilmReceive()

{

InitializeComponent();

}

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

{

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

}

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

{

Close();

}

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

{

историяTableAdapter.FillByAbsentUser(this.databaseDataSet.История, this.textBoxSearchUser.Text);

this.textBoxSearchName.Text = null;

this.textBoxSearchProducer.Text = null;

}

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

{

историяTableAdapter.FillByAbsentName(this.databaseDataSet.История, this.textBoxSearchName.Text);

this.textBoxSearchUser.Text = null;

this.textBoxSearchProducer.Text = null;

}

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

{

историяTableAdapter.FillByAbsentProducer(this.databaseDataSet.История, this.textBoxSearchProducer.Text);

this.textBoxSearchUser.Text = null;

this.textBoxSearchName.Text = null;

}

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

{

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

this.textBoxSearchUser.Text = null;

this.textBoxSearchName.Text = null;

this.textBoxSearchProducer.Text = null;

}

private void DataGridView\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

int codeUser = Convert.ToInt32(dataGridView.CurrentRow.Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString());

int codeFilm = Convert.ToInt32(dataGridView.CurrentRow.Cells["кодФильмаDataGridViewTextBoxColumn"].Value.ToString());

int codeRecord = Convert.ToInt32(dataGridView.CurrentRow.Cells["кодЗаписиDataGridViewTextBoxColumn"].Value.ToString());

string user = dataGridView.CurrentRow.Cells["пользовательDataGridViewTextBoxColumn"].Value.ToString();

string filmName = dataGridView.CurrentRow.Cells["фильмDataGridViewTextBoxColumn"].Value.ToString();

string filmProducer = dataGridView.CurrentRow.Cells["режиссерDataGridViewTextBoxColumn"].Value.ToString();

string filmYear = dataGridView.CurrentRow.Cells["годВыходаDataGridViewTextBoxColumn"].Value.ToString();

var condition = MessageBox.Show(

"Фильм: \n Название: \"" + filmName + "\"\n Режиссер(-ы): " + filmProducer + "\n Год выхода: " + filmYear + "\n Пользователь: " + user,

"Возвращение фильма в картотеку",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

try

{

фильмыTableAdapter.UpdateAbsence(false, codeFilm);

пользователиTableAdapter.UpdateLastVisit(DateTime.Now.Date, codeUser);

историяTableAdapter.UpdateReturnDate(DateTime.Now.Date, codeRecord);

MessageBox.Show(

"Действие успешно выполнено!",

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

}

catch (Exception)

{

MessageBox.Show(

"Произошла ошибка, попробуйте снова",

"Ошибка",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

return;

}

break;

}

}

}

}

**FormView.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace RGR

{

public partial class FormView : Form

{

public FormView()

{

InitializeComponent();

}

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

{

this.фильмыTableAdapter.Fill(this.databaseDataSet.Фильмы);

this.пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

this.историяTableAdapter.Fill(this.databaseDataSet.История);

this.panelUser.Dock = System.Windows.Forms.DockStyle.Fill;

this.panelFilm.Dock = System.Windows.Forms.DockStyle.Fill;

this.panelHistory.Dock = System.Windows.Forms.DockStyle.Fill;

this.dataGridViewFilm.Sort(this.dataGridViewFilm.Columns[названиеDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

this.dataGridViewUser.Sort(this.dataGridViewUser.Columns[фамилияDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

this.dataGridViewHistory.Sort(this.dataGridViewHistory.Columns[датаВзятияDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Смена окон \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

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

{

panelFilm.Show();

panelUser.Hide();

panelHistory.Hide();

}

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

{

panelFilm.Hide();

panelUser.Show();

panelHistory.Hide();

}

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

{

panelFilm.Hide();

panelUser.Hide();

panelHistory.Show();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно фильмов \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

private void RadioBtnFilm\_CheckedChanged(object sender, EventArgs e)

{

if (radioBtnFilmAll.Checked)

фильмыTableAdapter.Fill(this.databaseDataSet.Фильмы);

else if (radioBtnFilmPresent.Checked)

фильмыTableAdapter.FillByAbsent(this.databaseDataSet.Фильмы, false);

else if (radioBtnFilmAbsent.Checked)

фильмыTableAdapter.FillByAbsent(this.databaseDataSet.Фильмы, true);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

// Выделение цветом

private void DataGridViewFilm\_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)

{

foreach (DataGridViewRow row in dataGridViewFilm.Rows)

{

if ((bool)row.Cells["отсутствуетDataGridViewCheckBoxColumn"].Value)

dataGridViewFilm.Rows[row.Index].DefaultCellStyle.BackColor = Color.LightPink;

else

dataGridViewFilm.Rows[row.Index].DefaultCellStyle.BackColor = Color.LightGreen;

}

}

// Убирает выделение строк

private void DataGridViewFilm\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewFilm.ClearSelection();

}

// Поиск фильма

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

{

string btnName = (sender as Button).Name.ToString();

if (radioBtnFilmPresent.Checked)

{

var condition = MessageBox.Show(

"Поиск будет выполнен только по присутствующим фильмам",

"Предупреждение",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName2(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text, (bool)false);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre2(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text, (bool)false);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor2(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text, (bool)false);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

break;

}

}

else if (radioBtnFilmAbsent.Checked)

{

var condition = MessageBox.Show(

"Поиск будет выполнен только по отсутствующим фильмам",

"Предупреждение",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Warning);

switch (condition)

{

case DialogResult.Cancel: return;

case DialogResult.OK:

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName2(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text, (bool)true);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre2(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text, (bool)true);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor2(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text, (bool)true);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

break;

}

}

else

{

if (btnName == "buttonSearchName")

{

фильмыTableAdapter.FillByName(this.databaseDataSet.Фильмы, this.textBoxSearchName.Text);

this.textBoxSearchGenre.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchGenre")

{

фильмыTableAdapter.FillByGenre(this.databaseDataSet.Фильмы, this.textBoxSearchGenre.Text);

this.textBoxSearchName.Text = null;

this.textBoxSearchActor.Text = null;

}

else if (btnName == "buttonSearchActor")

{

фильмыTableAdapter.FillByActor(this.databaseDataSet.Фильмы, this.textBoxSearchActor.Text);

this.textBoxSearchName.Text = null;

this.textBoxSearchGenre.Text = null;

}

}

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно пользователей \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

// Поиск пользователя

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

{

пользователиTableAdapter.FillByFIO(this.databaseDataSet.Пользователи, this.textBoxSearchUserF.Text, this.textBoxSearchUserI.Text, this.textBoxSearchUserO.Text);

}

// Сброс поиска

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

{

пользователиTableAdapter.Fill(this.databaseDataSet.Пользователи);

this.textBoxSearchUserF.Text = null;

this.textBoxSearchUserI.Text = null;

this.textBoxSearchUserO.Text = null;

}

// Убирает выделение строк

private void DataGridViewUser\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewUser.ClearSelection();

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно истории \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

// Выделение цветом

private void DataGridViewHistory\_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)

{

foreach (DataGridViewRow row in dataGridViewHistory.Rows)

{

if (row.Cells["датаВозвратаDataGridViewTextBoxColumn"].Value.ToString() == "")

dataGridViewHistory.Rows[row.Index].DefaultCellStyle.BackColor = Color.LightPink;

else

dataGridViewHistory.Rows[row.Index].DefaultCellStyle.BackColor = Color.LightGreen;

}

}

// Убирает выделение строк

private void DataGridViewHistory\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewHistory.ClearSelection();

}

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

{

Close();

}

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

{

историяTableAdapter.FillByUser(this.databaseDataSet.История, this.textBoxSearchUser.Text);

this.textBoxSearchUser.Text = null;

this.textBoxSearchFilm.Text = null;

}

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

{

историяTableAdapter.FillByName(this.databaseDataSet.История, this.textBoxSearchFilm.Text);

this.textBoxSearchUser.Text = null;

this.textBoxSearchFilm.Text = null;

}

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

{

историяTableAdapter.Fill(this.databaseDataSet.История);

this.textBoxSearchUser.Text = null;

this.textBoxSearchFilm.Text = null;

}

}

}

**FormDebtors.cs**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using iTextSharp.text;

using iTextSharp.text.pdf;

namespace RGR

{

public partial class FormDebtors : Form

{

public FormDebtors()

{

InitializeComponent();

}

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

{

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

this.panelHistory.Dock = System.Windows.Forms.DockStyle.Fill;

this.dataGridViewHistory.Sort(this.dataGridViewHistory.Columns[пользовательDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно истории \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

private void DataGridViewHistory\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewHistory.ClearSelection();

}

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

{

this.dataGridViewHistory.Sort(this.dataGridViewHistory.Columns[пользовательDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

SaveFileDialog dlg = new SaveFileDialog();

dlg.FileName = "Должники " + DateTime.Now.ToShortDateString();

dlg.Filter = "PDF files (\*.pdf)|\*.pdf";

dlg.RestoreDirectory = true;

dlg.Title = "Сохранение файла";

if (dlg.ShowDialog() == DialogResult.OK)

{

try

{

string outputFile = dlg.FileName;

FileStream fs = new FileStream(outputFile, FileMode.Create, FileAccess.Write, FileShare.None);

Document doc = new Document(PageSize.A4, 25, 25, 25, 15);

PdfWriter writer = PdfWriter.GetInstance(doc, fs);

BaseFont baseFont = BaseFont.CreateFont(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "ARIAL.TTF"), BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);

iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.NORMAL);

doc.Open();

Paragraph p = new Paragraph("Список должников | " + DateTime.Now.ToShortDateString(), new iTextSharp.text.Font(baseFont, 16, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.SpacingAfter = 20;

p.Alignment = 1;

doc.Add(p);

if (dataGridViewHistory.Rows.Count == 0)

{

string text = "Должники отсутствуют";

p = new Paragraph(text, new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.IndentationLeft = 20;

p.IndentationRight = 20;

p.Alignment = 0;

doc.Add(p);

}

else

{

string text = "Список должников по состоянию на " + DateTime.Now.ToShortDateString() + ":";

p = new Paragraph(text, new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.SpacingAfter = 10;

p.IndentationLeft = 20;

p.Alignment = 0;

doc.Add(p);

iTextSharp.text.List listMain = new iTextSharp.text.List(List.UNORDERED);

listMain.IndentationLeft = 30;

listMain.SetListSymbol("\u2022");

DataGridView dgv = dataGridViewHistory;

while (dgv.Rows.Count != 0)

{

Int32 userID = Int32.Parse(dgv.Rows[0].Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString());

string userInitials = dgv.Rows[0].Cells["пользовательDataGridViewTextBoxColumn"].Value.ToString();

listMain.Add(new iTextSharp.text.ListItem(" " + userInitials+":", font));

string tempCode = dgv.Rows[0].Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString();

iTextSharp.text.List list = new iTextSharp.text.List(List.UNORDERED);

list.IndentationLeft = 30;

list.SetListSymbol("\u2012");

while ((dgv.Rows.Count != 0) && (tempCode == dgv.Rows[0].Cells["кодПользователяDataGridViewTextBoxColumn"].Value.ToString()))

{

DataGridViewRow row = dgv.Rows[0];

string filmName = row.Cells["фильмDataGridViewTextBoxColumn"].Value.ToString();

string filmYear = row.Cells["годВыходаDataGridViewTextBoxColumn"].Value.ToString();

string dateTake = row.Cells["датаВзятияDataGridViewTextBoxColumn"].Value.ToString().Split(' ')[0];

list.Add(new iTextSharp.text.ListItem(" \"" + filmName + "\" (" + filmYear + "); дата взятия: " + dateTake + ";", font));

dgv.Rows.RemoveAt(0);

}

listMain.Add(list);

}

doc.Add(listMain);

}

doc.Close();

writer.Close();

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

MessageBox.Show(

"Документ успешно создан! \n\n Файл сохранен по следующей директории: " + outputFile,

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

catch

{

MessageBox.Show(

"Ошибка! \n Не удалось сохранить файл! \n Проверьте права на запись в дирректории для сохранения и попробуйте снова!",

"Упс! Что-то пошло не так...",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

}

dlg.Dispose();

}

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

{

Close();

}

}

}

**FormMissingFilms.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using iTextSharp.text;

using iTextSharp.text.pdf;

namespace RGR

{

public partial class FormMissingFilms : Form

{

public FormMissingFilms()

{

InitializeComponent();

}

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

{

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

this.panelHistory.Dock = System.Windows.Forms.DockStyle.Fill;

this.dataGridViewHistory.Sort(this.dataGridViewHistory.Columns[фильмDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

}

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

/\* Окно истории \*/

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

private void DataGridViewHistory\_SelectionChanged(object sender, EventArgs e)

{

this.dataGridViewHistory.ClearSelection();

}

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

{

this.dataGridViewHistory.Sort(this.dataGridViewHistory.Columns[фильмDataGridViewTextBoxColumn.Index], ListSortDirection.Ascending);

SaveFileDialog dlg = new SaveFileDialog();

dlg.FileName = "Отсутствующие фильмы " + DateTime.Now.ToShortDateString();

dlg.Filter = "PDF files (\*.pdf)|\*.pdf";

dlg.RestoreDirectory = true;

dlg.Title = "Сохранение файла";

if (dlg.ShowDialog() == DialogResult.OK)

{

try

{

string outputFile = dlg.FileName;

FileStream fs = new FileStream(outputFile, FileMode.Create, FileAccess.Write, FileShare.None);

Document doc = new Document(PageSize.A4, 25, 25, 25, 15);

PdfWriter writer = PdfWriter.GetInstance(doc, fs);

BaseFont baseFont = BaseFont.CreateFont(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "ARIAL.TTF"), BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);

iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.NORMAL);

doc.Open();

Paragraph p = new Paragraph("Отсутствующие фильмы", new iTextSharp.text.Font(baseFont, 16, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.SpacingAfter = 20;

p.Alignment = 1;

doc.Add(p);

if (dataGridViewHistory.Rows.Count == 0)

{

string text = "По состоянию на "+ DateTime.Now.ToShortDateString() + ", все фильмы находятся в фильмотеку";

p = new Paragraph(text, new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.IndentationLeft = 20;

p.IndentationRight = 20;

p.Alignment = 0;

doc.Add(p);

}

else

{

string text = "По состоянию на " + DateTime.Now.ToShortDateString() + ", отсутствуют следующие фильмы:";

p = new Paragraph(text, new iTextSharp.text.Font(baseFont, 14, iTextSharp.text.Font.BOLD));

p.SpacingBefore = 20;

p.SpacingAfter = 10;

p.IndentationLeft = 20;

p.Alignment = 0;

doc.Add(p);

iTextSharp.text.List listMain = new iTextSharp.text.List(List.UNORDERED);

listMain.IndentationLeft = 30;

listMain.SetListSymbol("\u2022");

DataGridView dgv = dataGridViewHistory;

for (int i = 0; i < dgv.Rows.Count; i++)

{

string user = dgv.Rows[i].Cells["пользовательDataGridViewTextBoxColumn"].Value.ToString();

string filmName = dgv.Rows[i].Cells["фильмDataGridViewTextBoxColumn"].Value.ToString();

string filmCountry = dgv.Rows[i].Cells["странаDataGridViewTextBoxColumn"].Value.ToString();

string filmProducer = dgv.Rows[i].Cells["режиссерDataGridViewTextBoxColumn"].Value.ToString();

string filmYear = dgv.Rows[i].Cells["годВыходаDataGridViewTextBoxColumn"].Value.ToString();

string dateTake = dgv.Rows[i].Cells["датаВзятияDataGridViewTextBoxColumn"].Value.ToString().Split(' ')[0];

listMain.Add(new iTextSharp.text.ListItem(" \"" + filmName + "\":", font));

iTextSharp.text.List list = new iTextSharp.text.List(List.UNORDERED);

list.IndentationLeft = 30;

list.SetListSymbol("\u2012");

list.Add(new iTextSharp.text.ListItem(" Год выхода: " + filmYear, font));

list.Add(new iTextSharp.text.ListItem(" Страна: " + filmCountry, font));

list.Add(new iTextSharp.text.ListItem(" Режиссер: " + filmProducer, font));

list.Add(new iTextSharp.text.ListItem(" На руках у: " + user, font));

list.Add(new iTextSharp.text.ListItem(" Дата взятия: " + dateTake, font));

listMain.Add(list);

}

doc.Add(listMain);

}

doc.Close();

writer.Close();

историяTableAdapter.FillByAbsent(this.databaseDataSet.История);

MessageBox.Show(

"Документ успешно создан! \n\n Файл сохранен по следующей директории: " + outputFile,

"Уведомление",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

catch

{

MessageBox.Show(

"Ошибка! \n Не удалось сохранить файл! \n Проверьте права на запись в дирректории для сохранения и попробуйте снова!",

"Упс! Что-то пошло не так...",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

}

dlg.Dispose();

}

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

{

Close();

}

}

}