Федеральное государственное образовательное учреждение

высшего образования

«**КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**»

Кафедра систем управления и вычислительной техники

|  |  |
| --- | --- |
| Курсовая работа  допущена к защите  руководитель доцент Высоцкий Л. Г. | Курсовая работа защищена  с оценкой …………………  руководитель доцент Высоцкий Л. Г. |

**Программная система «Приемная комиссия»**

Курсовая работа по дисциплине

“Разработка и стандартизация программных средств и информационных технологий”

Пояснительная записка

Нормоконтроллер Проект выполнила

Высоцкий Л. Г. студентка гр. 17 – ВТ

Камалова К.П.

Калининград

2020

**Оглавление**

[Введение 3](#_Toc50637491)

[1. Спецификация программной системы 4](#_Toc50637492)

[1.1. Дерево функций проекта 4](#_Toc50637493)

[1.2. Дерево модулей проекта 5](#_Toc50637494)

[1.3. Распределение функций системы по модулям 5](#_Toc50637495)

[1.4. Структуры используемых данных 5](#_Toc50637496)

[1.5. Видеоформы пользовательского интерфейса 6](#_Toc50637497)

[2. Руководство системного программиста 8](#_Toc50637498)

[2.1 Сведения о системе. 8](#_Toc50637499)

[2.2. Установка программы. 8](#_Toc50637500)

[2.3. Удаление программы. 8](#_Toc50637501)

[3. Руководство пользователя 8](#_Toc50637502)

[3.1. Общие сведения о программе 8](#_Toc50637503)

[3.2. Установка 9](#_Toc50637504)

[3.3.Запуск 9](#_Toc50637505)

[3.4. Инструкция по работе 9](#_Toc50637506)

[3.5. Сообщения пользователю 10](#_Toc50637507)

[4. Справочная система 12](#_Toc50637508)

[5. Листинги модулей 18](#_Toc50637509)

[5.1 Модуль LogInForm 18](#_Toc50637510)

[5.2 Модуль MainForm 23](#_Toc50637511)

[5.3 Модуль AddAndModForm 28](#_Toc50637512)

[Используемая литература 30](#_Toc50637513)

# Введение

Представленная в данной работе программная система «Картотека работников» является реализацией задания на выполнение курсовой работы по дисциплине «Разработка программных комплексов», задание сформулировано следующим образом: Информационная система «Картотека работников». Занесение данных о работниках на предприятии.

Также у пользователя есть возможность узнать сведения об авторе данной программной системы и воспользоваться средствами справочной системы.

# 1. Спецификация программной системы

## 1.1. Дерево функций проекта

Все действия (функции) программного продукта можно разделить на подмножества функций (см. рис. 1).

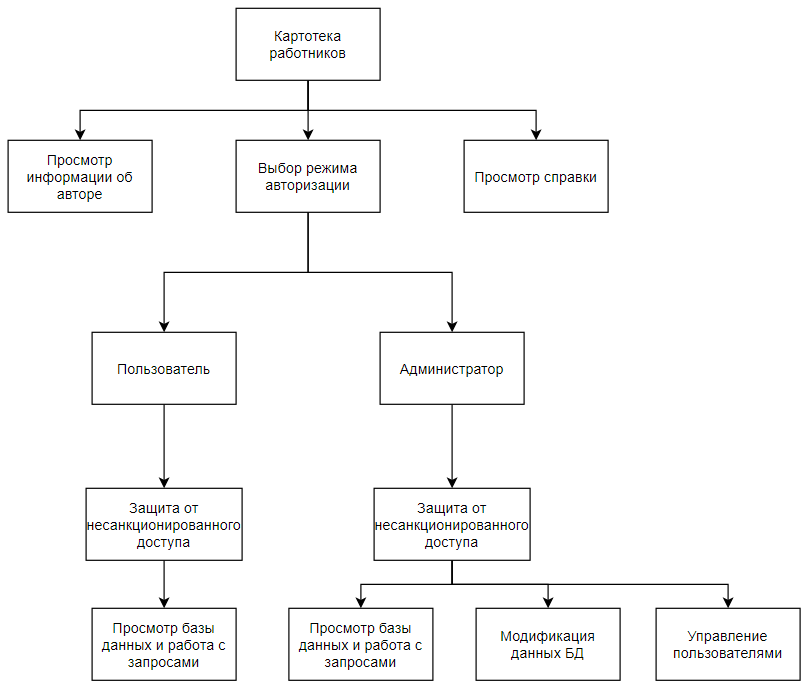


Рис. 1 Дерево функций проекта

## 1.2. Дерево модулей проекта

На рис. 2 представлено дерево модулей данной системы.

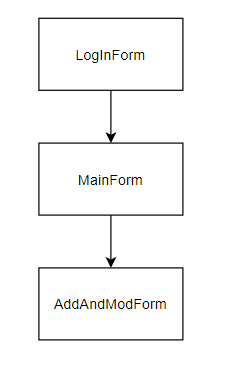


Рис. 2 Дерево модулей

## 1.3. Распределение функций системы по модулям

**Модуль LogInForm**

Включает в себя выбор режима авторизации, защиту от несанкционированного доступа, вывод информации об авторе, для администратора добавление и удаление пользователей

**Модуль MainForm**

Включает в себя отображение содержимого базы данных, запросы для базы данных, если авторизован администратор, то добавление, удаление и модификация данных о работниках

**Модуль AddAndModForm**

Форма для добавления работника или модификации данных о уже существующем работнике

## 1.4. Структуры используемых данных

|  |  |  |
| --- | --- | --- |
| Данные | Тип | Применение |
| ConnStr | string | Переменная, котороя определяет строку подключения к базе данных |
| fileName | String | Переменная, содержащая путь до файла с паролями пользователей |
| IsAdm | integer | Переменная, которая определяет является ли пользователь администратором |
| Key | String | Переменные, обозначающие ключ для шифрования пароля и даты |
| i | integer | Переменная, обозначающая количество строк в таблице для работы с ней, как с массивом |
| j | integer | Переменная, обозначающая количество столбцов в таблице для работы с ней, как с массивом |
| Users  Name  Password | Class  String  String | Переменная, которой присваивается имя пользователя и пароль |

На рис.3 представлена база данных «Workers.mdb». Имеющая следующую структуру.

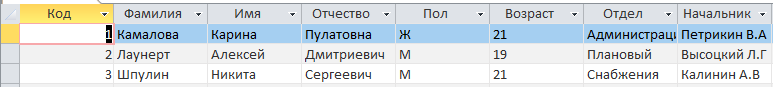


Рис. 3 Структура базы данных Workers.mdb

## 1.5. Видеоформы пользовательского интерфейса

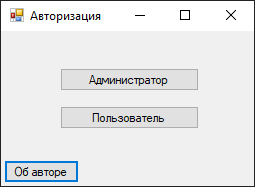


Рис.4 Видеоформа окна Авторизации(модуль LogInForm)

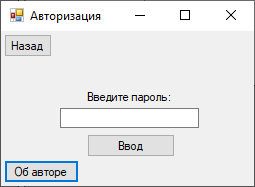


Рис.5 Видеоформа защиты от НСД при авторизации администратора (модуль LogInForm)

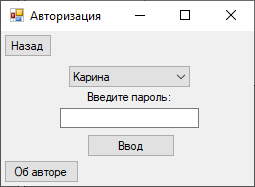


Рис.6 Видеоформа защиты от НСД при авторизации Рядового пользователя (модуль LogInForm)

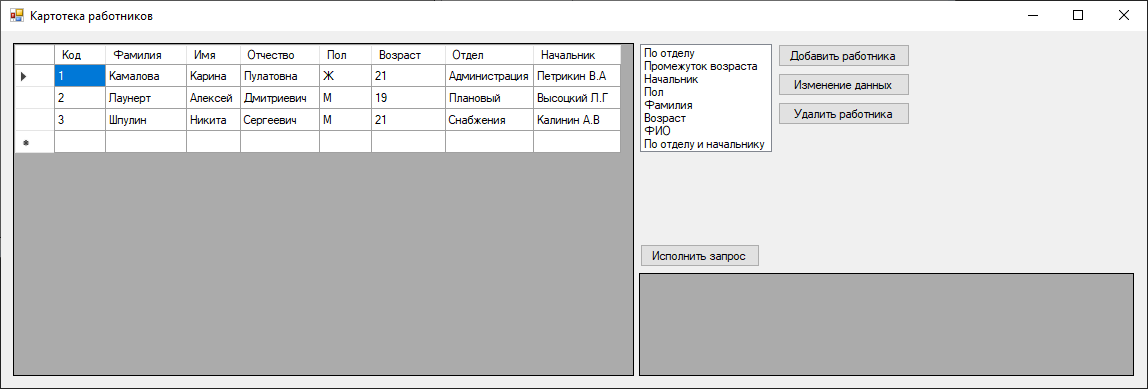


Рис. 7 Видеоформа Главного окна при авторизации администратора (модуль MainForm)

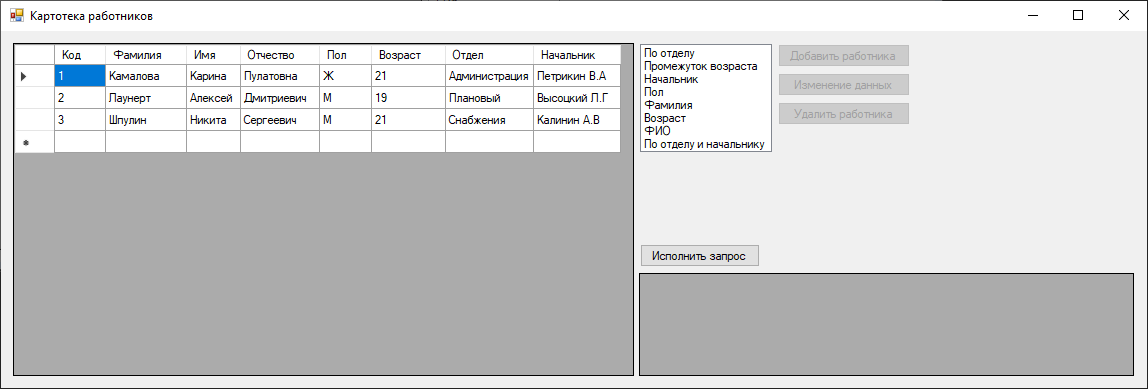


Рис.8 Видеоформа Главного окна при авторизации рядового пользователя (модуль MainForm)

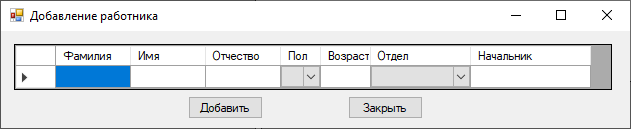


Рис.9 Видеоформа окна добавления Работника (модуль AddAndModForm)

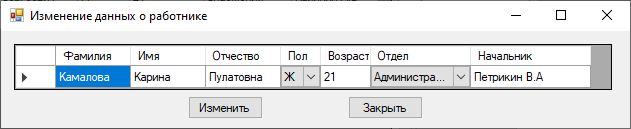


Рис.10 Видеоформа окна Изменения данных о Работнике (модуль AddAndModForm)

# 2. Руководство системного программиста

## 2.1 Сведения о системе.

Программная система «Картотека работников» предназначена для работы в двух режимах : Пользователь и Администратор.

«Пользователь» может просматривать картотеку, исполнять запросы, которые выводят списки работников подходящих под критерии запроса.

«Администратор» имеет расширенный функционал работы с базой данных. Он может добавлять, изменять и модифицировать информацию о работниках.

## 2.2. Установка программы.

Для установки ПС необходимо разархивировать архив “Картотека работников.rar” в необходимую папку.

## 2.3. Удаление программы.

Для полного удаления программы необходимо удалить папку «Release» со всем содержимым.

# 3. Руководство пользователя

## 3.1. Общие сведения о программе

Программная система «Картотека работников» предназначена для работы в двух режимах : Пользователь и Администратор.

«Пользователь» может просматривать картотеку, исполнять запросы, которые выводят списки работников, подходящих под критерии запроса.

«Администратор» имеет расширенный функционал работы с базой данных. Он может добавлять, изменять и модифицировать информацию о работниках.

## 3.2. Установка

Для установки ПС необходимо разархивировать архив “Картотека работников.rar” в необходимую папку.

## 3.3.Запуск

Для запуска программной системы необходимо запустить исполняемый файл Kurs\_RPK.exe из ранее установленной директории Release.

## 3.4. Инструкция по работе

В данном разделе описана работа с программной системой в двух режимах: Пользователь и Администратор.

Работа в режиме пользователя состоит из просмотра с данными о работниках предприятия. Также пользователь может исполнять запросы базы данных.

Работа в режиме Администратор заключается в редактировании базы данных c информацией о работниках. Администратор может добавлять и удалять работников, а также изменять информацию о них

При добавлении работника или изменении информации о работнике, Администратор должен помнить, что все поля обязательны для заполнения, Администратор получит предупреждение системы в том случае, если одно из полей останется незаполненным.

## 3.5. Сообщения пользователю

|  |  |
| --- | --- |
| **Сообщение системы** | **Пояснение** |
|  | Данное сообщение появляется при нажатии на кнопку «Об авторе» |
|  | Данное сообщение появляется при добавлении пользователя для авторизации в систему |
|  | Данное сообщение появляется при удалении пользователя |
|  | Данное сообщение появляется, когда пользователь успешно авторизуется |
|  | Данное сообщение появляется если при авторизации не был выбран пользователь |
|  | Данное сообщение появляется если при авторизации был введен неверный пароль |
|  | Данное сообщение появляется, когда администратор добавляет работника в базу данных |
|  | Данное сообщение появляется, когда администратор изменяет данные о работнике |
|  | Данное сообщение появляется в случае, когда администратор не заполнил все поля при добавлении работника или изменении данных о нем |
|  | Данное сообщение появляется, когда администратор удаляет работника |

# 4. Справочная система

Программная система «Приемная комиссия» оснащена справочной системой, начальная страница и содержание которой представлены на рис.9.

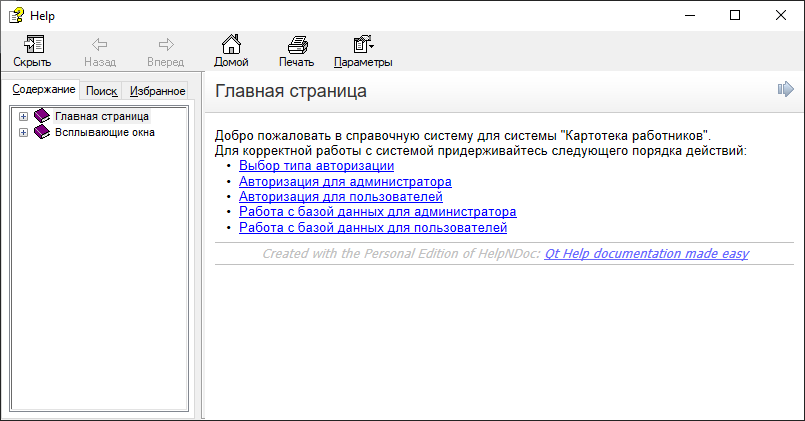


Рис. 11 Начальная страница справочной системы

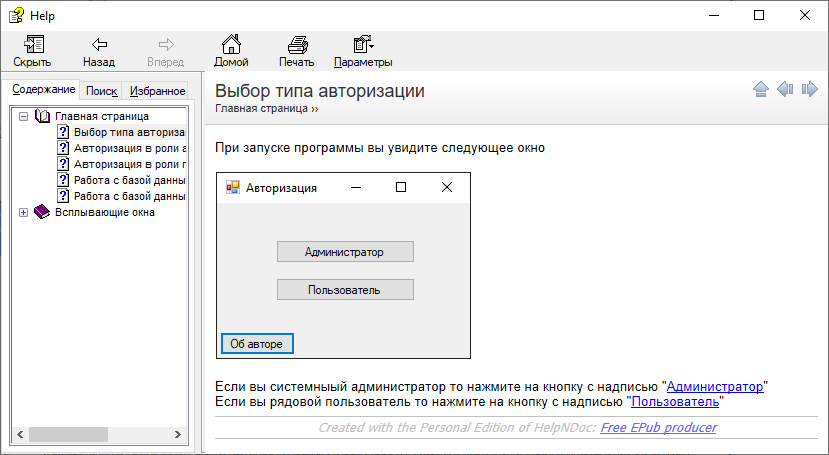


Рис. 12 Авторизация

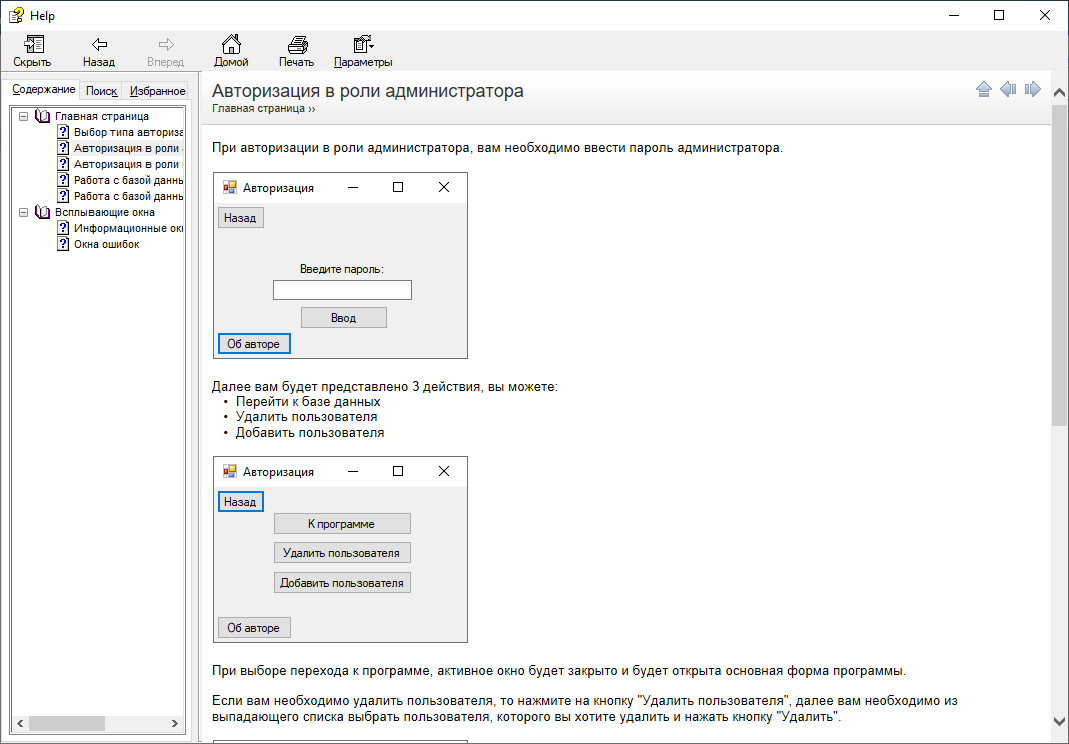


Рис. 13 Авторизация в роли администратора

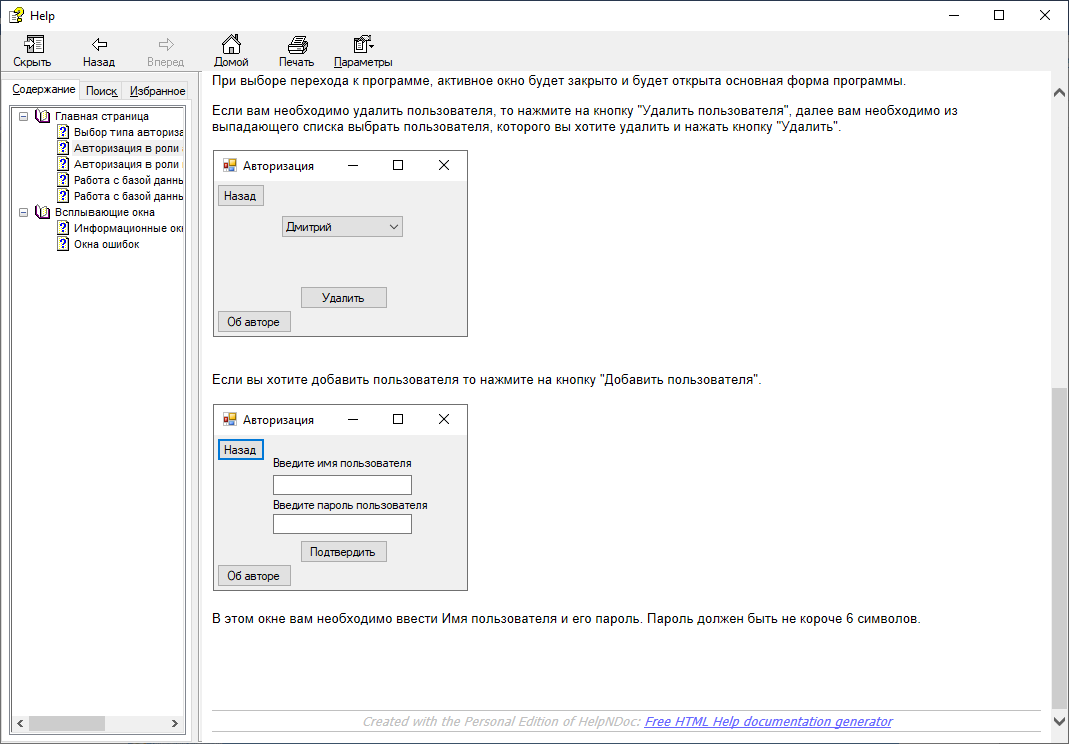


Рис. 14 Добавление и удаление пользователей

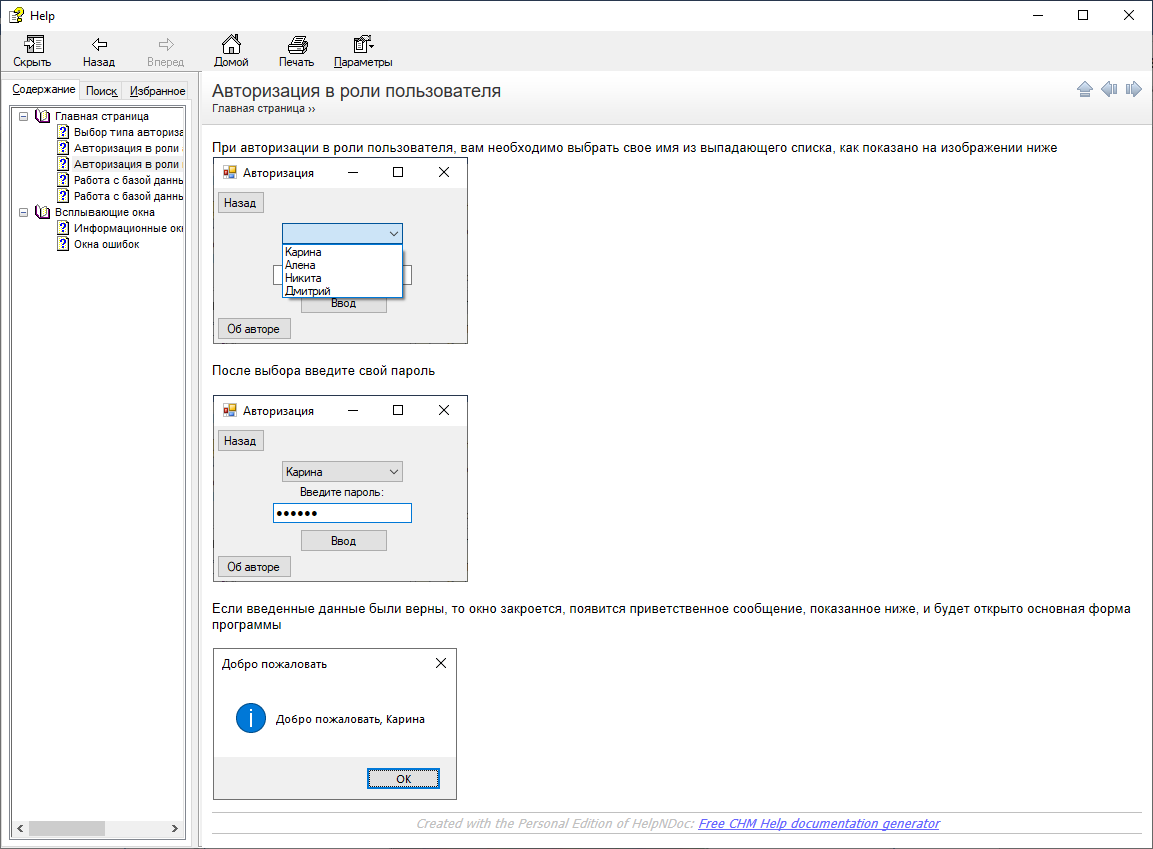


Рис. 15 Авторизация в роли пользователя

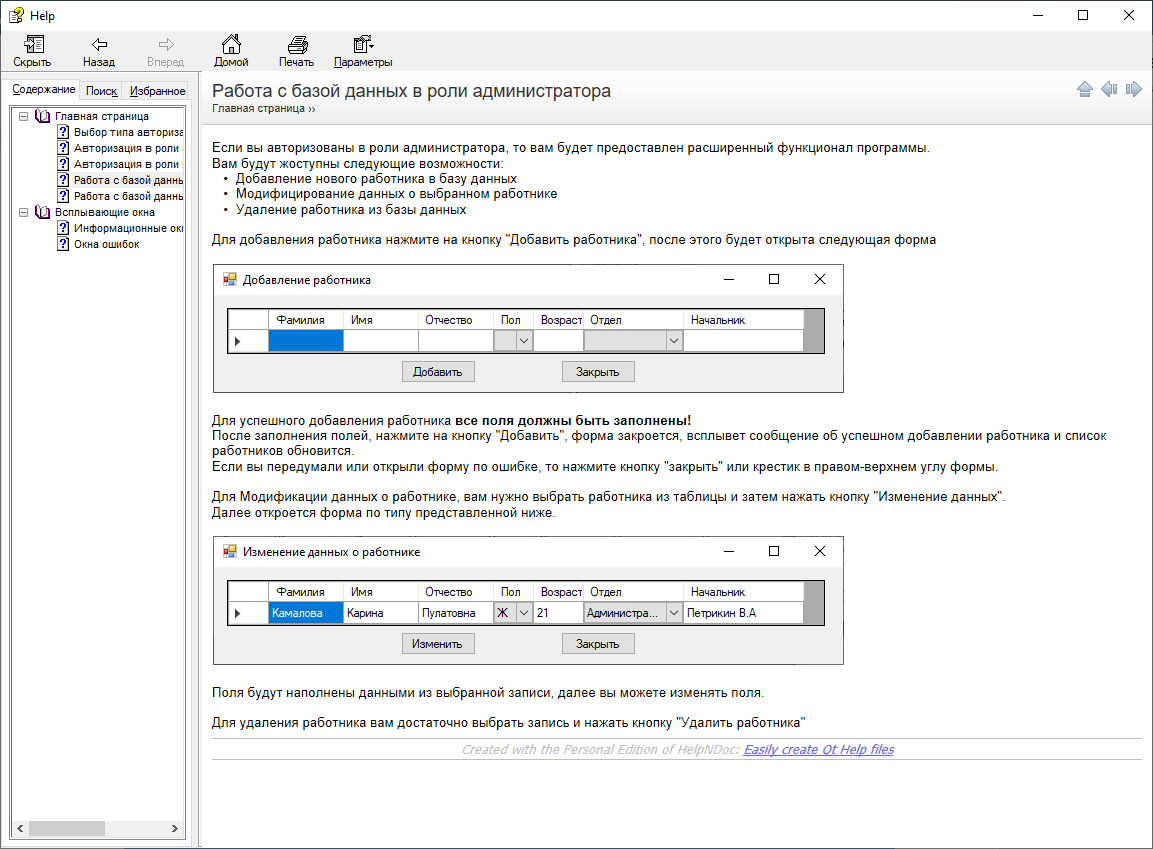


Рис. 16 Расширенные возможности администратора

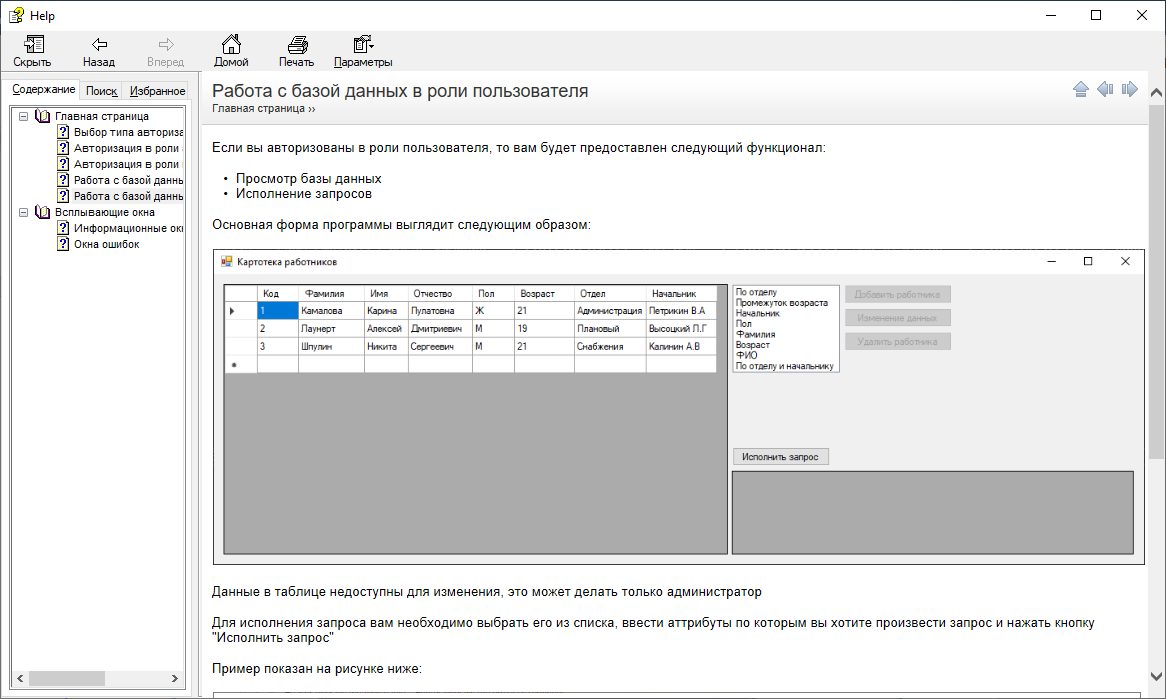


Рис. 17 Работа с базой данных в роли пользователя

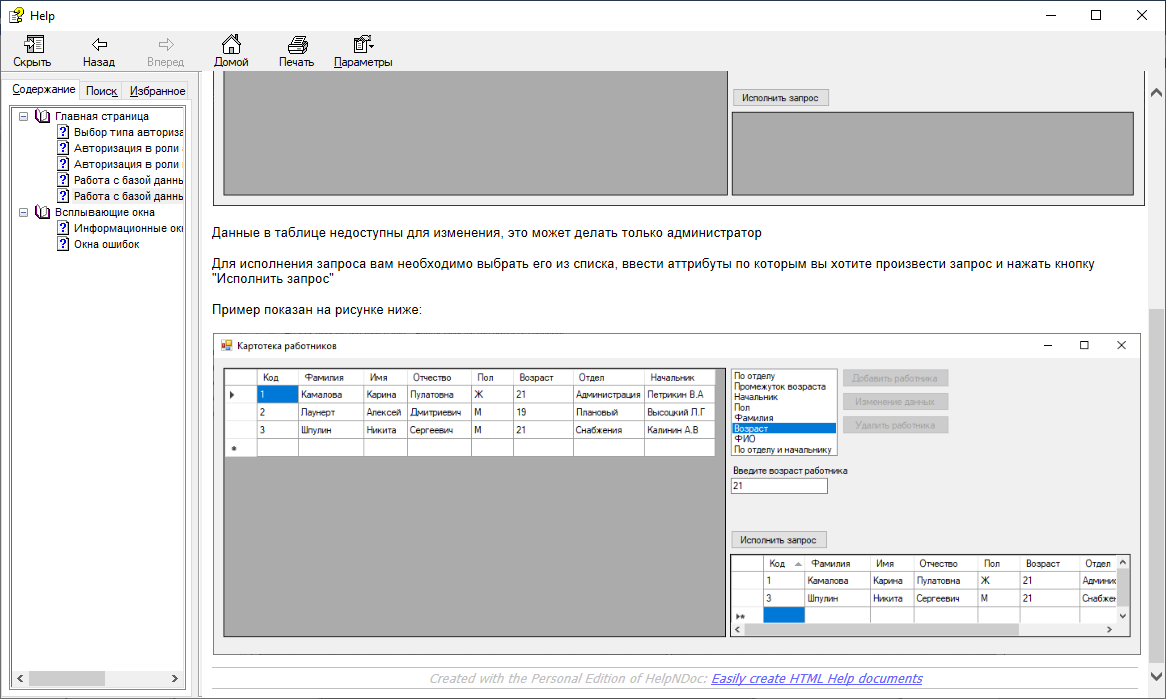


Рис. 18 Пример исполнения запроса

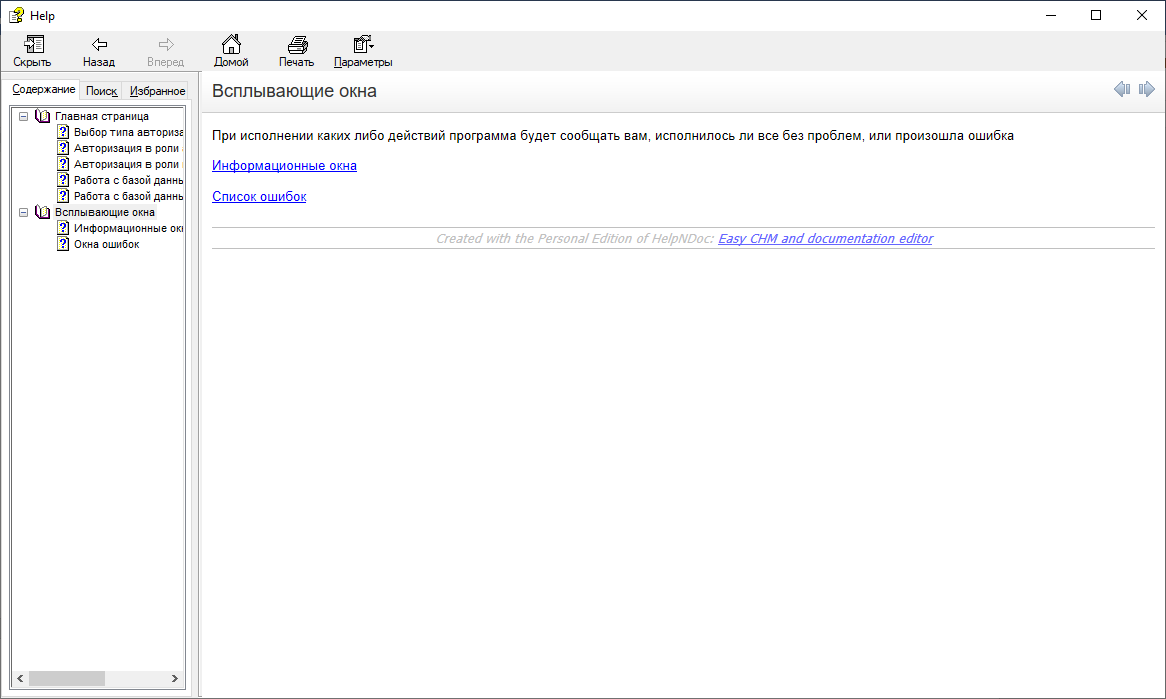


Рис. 19 Краткая информация о всплывающих окнах

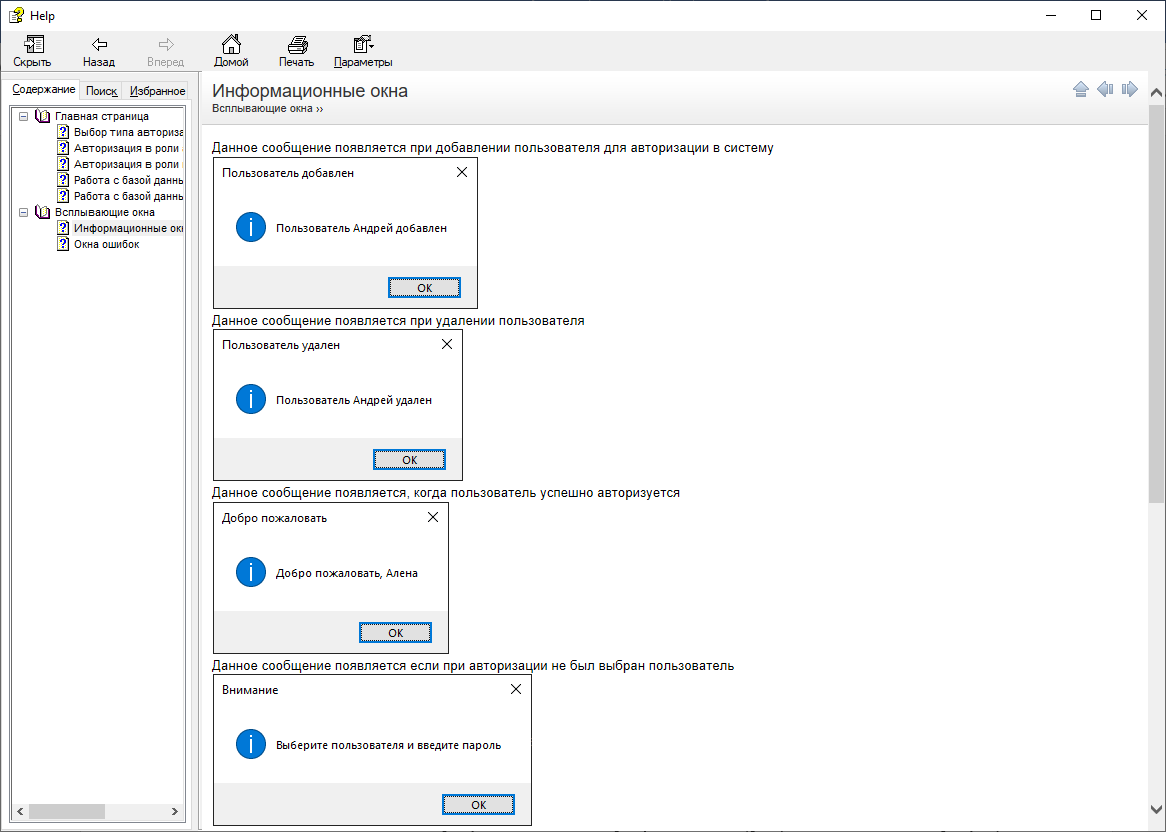


Рис. 20 Информационные сообщения

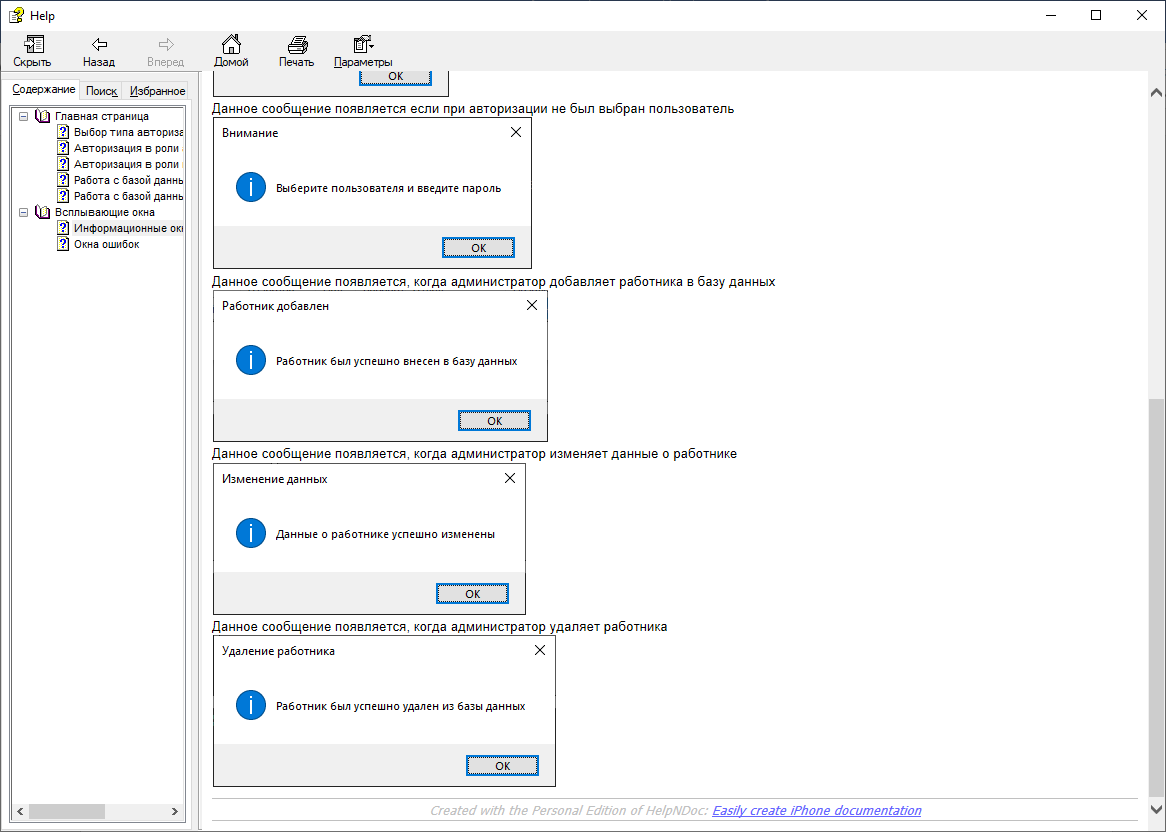


Рис. 21 информационные сообщения

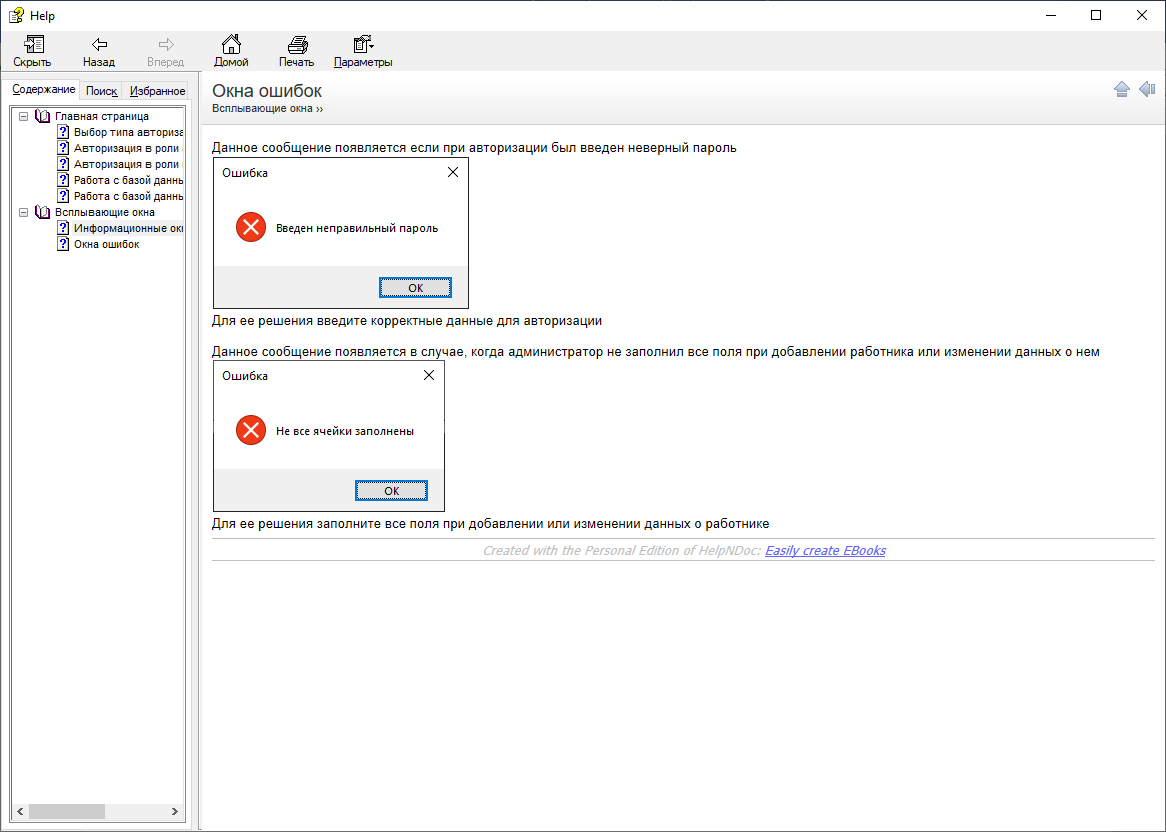


Рис.22 Сообщения об ошибках

# 5. Листинги модулей

## 5.1 Модуль LogInForm

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows.Forms;

namespace Kurs\_RPK

{

public partial class LogInForm : Form

{

// Пароль администратора

readonly string AdmPass = "Administrator";

public bool isAdm = false;

//Класс пользователя

public class Users

{

public string Name { get; set; }

public string Password { get; set; }

}

//Список пользователей

List<Users> User = new List<Users> { };

public LogInForm()

{

InitializeComponent();

Program.f1 = this;

}

//Метод добавления пользователей

private void UserAdd()

{

string fileName = "Users.txt";

using (StreamWriter writer = new StreamWriter(fileName, true))

{

if ((AddUserName.Text != "") && (AddUserPass.Text != ""))

{

if(AddUserPass.Text.Length >= 6)

{

writer.WriteLine(AddUserName.Text + " " + AddUserPass.Text);

MessageBox.Show($"Пользователь {AddUserName.Text} добавлен", "Пользователь добавлен", MessageBoxButtons.OK,MessageBoxIcon.Information);

}

else MessageBox.Show("Пароль не может быть короче 6 символов", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else MessageBox.Show("Одно из полей не заполнено","Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

UserList.Items.Clear();

User.Clear();

FillList();

}

//Метод удаления пользователей

private void UserDelete()

{

string tempFile = "Temp.txt";

int index = User.FindIndex(x => x.Name.Contains(UserList.Text));

string toDelete = User[index].Name + " " + User[index].Password;

if (UserList.SelectedIndex != -1)

{

using (var sr = new StreamReader("Users.txt"))

using (var sw = new StreamWriter(tempFile))

{

string line;

while ((line = sr.ReadLine()) != null)

{

if (line != toDelete)

{

sw.WriteLine(line);

}

}

}

File.Delete("Users.txt");

File.Move(tempFile, "Users.txt");

MessageBox.Show($"Пользователь {User[index].Name} удален", "Пользователь удален", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else MessageBox.Show("Выберите пользователся из списка","Не выбран пользователь",MessageBoxButtons.OK, MessageBoxIcon.Error);

UserList.SelectedIndex = -1;

UserList.Items.Clear();

User.Clear();

FillList();

}

//Метод заполнения списка пользователей

private void FillList()

{

string fileName = "Users.txt";

using (StreamReader reader = new StreamReader(fileName))

{

string user;

while ((user = reader.ReadLine()) != null){

string[] info = user.Split(new char[] { ' ' });

User.Add(new Users() { Name = info[0], Password = info[1] });

}

}

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

{

UserList.Items.Add(User[i].Name);

}

}

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

{

FillList();

}

//события на нажатия кнопок

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

{

AdminB.Visible = false;

UserB.Visible = false;

PasswordTB.Visible = true;

EnterPassLab.Visible = true;

PassEnter.Visible = true;

isAdm = true;

BackB.Visible = true;

}

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

{

AdminB.Visible = false;

UserB.Visible = false;

PasswordTB.Visible = true;

EnterPassLab.Visible = true;

PassEnter.Visible = true;

UserList.Visible = true;

BackB.Visible = true;

}

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

{

UserAdd();

EnterPWL.Visible = false;

EnterUNL.Visible = false;

AddUserB.Visible = false;

AddUserName.Text = "";

AddUserName.Visible = false;

AddUserPass.Text = "";

AddUserPass.Visible = false;

Begin.Visible = true;

AddUser.Visible = true;

DeleteUser.Visible = true;

}

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

{

EnterPWL.Visible = true;

EnterUNL.Visible = true;

Begin.Visible = false;

DeleteUser.Visible = false;

AddUser.Visible = false;

AddUserName.Visible = true;

AddUserPass.Visible = true;

AddUserB.Visible = true;

}

//Проверка паролей

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

{

if (isAdm == true)

{

if (PasswordTB.Text == AdmPass)

{

PasswordTB.Visible = false;

EnterPassLab.Visible = false;

PassEnter.Visible = false;

Begin.Visible = true;

DeleteUser.Visible = true;

AddUser.Visible = true;

PasswordTB.Text = "";

}

else MessageBox.Show("Введен неправильный пароль","Ошибка",MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

if (UserList.SelectedIndex != -1)

{

if (PasswordTB.Text == User[UserList.SelectedIndex].Password)

{

MessageBox.Show("Добро пожаловать, " + User[UserList.SelectedIndex].Name, "Добро пожаловать", MessageBoxButtons.OK, MessageBoxIcon.Information);

Program.f2 = new MainForm();

Program.f2.Show();

this.Hide();

}

else MessageBox.Show("Введен неправильный пароль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else MessageBox.Show("Выберите пользователя и введите пароль", "Внимание", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

Begin.Visible = false;

DeleteUser.Visible = false;

AddUser.Visible = false;

UserList.Visible = true;

DelUser.Visible = true;

}

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

{

UserDelete();

DelUser.Visible = false;

UserList.Visible = false;

Begin.Visible = true;

AddUser.Visible = true;

DeleteUser.Visible = true;

}

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

{

if (PasswordTB.Visible == true)

{

AdminB.Visible = true;

UserB.Visible = true;

PasswordTB.Visible = false;

EnterPassLab.Visible = false;

PassEnter.Visible = false;

isAdm = false;

UserList.Visible = false;

BackB.Visible = false;

}

else if (Begin.Visible == true)

{

PasswordTB.Visible = true;

EnterPassLab.Visible = true;

PassEnter.Visible = true;

Begin.Visible = false;

DeleteUser.Visible = false;

AddUser.Visible = false;

}

else if (DelUser.Visible == true || AddUserB.Visible == true)

{

EnterPWL.Visible = false;

EnterUNL.Visible = false;

AddUserB.Visible = false;

AddUserName.Text = "";

AddUserName.Visible = false;

AddUserPass.Text = "";

AddUserPass.Visible = false;

DelUser.Visible = false;

UserList.Visible = false;

Begin.Visible = true;

AddUser.Visible = true;

DeleteUser.Visible = true;

}

}

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

{

MessageBox.Show("Система разработана студенткой 17-ВТ-1\nКамаловой Кариной Пулатовной", "Об авторе", MessageBoxButtons.OK);

}

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

{

Program.f2 = new MainForm();

Program.f2.Show();

this.Hide();

}

}

}

## 5.2 Модуль MainForm

using System;

using System.Windows.Forms;

namespace Kurs\_RPK {

public partial class MainForm : Form

{

Query controller;

public MainForm()

{

InitializeComponent();

//Проверка, является ли пользователь администратором

if (Program.f1.isAdm)

{

this.ModifyEmployee.Enabled = true;

this.AddEmployee.Enabled = true;

this.DeleteEmployee.Enabled = true;

}

else

{

this.ModifyEmployee.Enabled = false;

this.AddEmployee.Enabled = false;

this.DeleteEmployee.Enabled = false;

}

controller = new Query(ConnectionStr.ConnStr);

//Заполнение таблицы данными из базы данных

dataGridView1.DataSource = controller.UpdateEmployee();

}

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

{

AddAndModForm modify = new AddAndModForm();

modify.Text = "Изменение данных о работнике";

modify.Show();

this.Enabled = false;

}

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

{

AddAndModForm add =new AddAndModForm();

add.Text = "Добавление работника";

add.Show();

this.Enabled=false;

}

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

{

controller.Delete(Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value.ToString()));

MessageBox.Show("Работник был успешно удален из базы данных","Удаление работника",MessageBoxButtons.OK,MessageBoxIcon.Information);

dataGridView1.DataSource = controller.UpdateEmployee();

}

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

{

Application.Exit();

}

//отображение окон ввода, соответствующих выбранному запросу

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

{

switch (QueryList.SelectedIndex)

{

case 0:

{

QueryCBL.Visible = true;

QueryCBL.Text = "Выберите отдел";

QueryMTB.Visible = false;

QueryTB.Visible = false;

QueryTBL.Visible = false;

QueryCB.Visible = true;

QueryCB.Items.Clear();

QueryCB.Items.AddRange(new object[] {

"Бухгалтерия",

"Кадров",

"Плановый",

"Снабжения",

"Администрация"});

break;

}

case 1:

{

QueryCBL.Visible = true;

QueryCBL.Text = "Выберите промежуток возраста";

QueryMTB.Visible = false;

QueryTB.Visible = false;

QueryTBL.Visible = false;

QueryCB.Visible = true;

QueryCB.Items.Clear();

QueryCB.Items.AddRange(new object[] {

"20 - 30",

"30 - 40",

"40 - 50",

"50 - 60",

"60 - 70"});

break;

}

case 2:

{

QueryTBL.Visible = true;

QueryTBL.Text = "Введите фамилию начальника";

QueryMTB.Visible = false;

QueryCB.Visible = false;

QueryCBL.Visible = false;

QueryTB.Visible = true;

break;

}

case 3:

{

QueryCBL.Visible = true;

QueryCBL.Text = "Выберите пол";

QueryMTB.Visible = false;

QueryTB.Visible = false;

QueryTBL.Visible = false;

QueryCB.Visible = true;

QueryCB.Items.Clear();

QueryCB.Items.AddRange(new object[] {

"М",

"Ж",});

break;

}

case 4:

{

QueryTBL.Visible = true;

QueryTBL.Text = "Введите фамилию работника";

QueryMTB.Visible = false;

QueryCB.Visible = false;

QueryCBL.Visible = false;

QueryTB.Visible = true;

break;

}

case 5:

{

QueryTBL.Visible = true;

QueryTBL.Text = "Введите возраст работника";

QueryMTB.Visible = true;

QueryCB.Visible = false;

QueryCBL.Visible = false;

QueryTB.Visible = false;

break;

}

case 6:

{

QueryTBL.Visible = true;

QueryTBL.Text = "Введите ФИО работника";

QueryMTB.Visible = false;

QueryCB.Visible = false;

QueryCBL.Visible = false;

QueryTB.Visible = true;

break;

}

case 7:

{

QueryCBL.Visible = true;

QueryCBL.Text = "Выберите отдел";

QueryCB.Items.Clear();

QueryCB.Items.AddRange(new object[] {

"Бухгалтерия",

"Кадров",

"Плановый",

"Снабжения",

"Администрация"});

QueryTBL.Visible = true;

QueryTBL.Text = "Введите фамилию начальника";

QueryCB.Visible = true;

QueryTB.Visible = true;

QueryMTB.Visible = false;

break;

}

}

}

//определение запроса, происходящего по нажатию кнопки "Исполнить запрос"

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

{

switch (QueryList.SelectedIndex)

{

case 0:

{

dataGridView2.DataSource = controller.DepartmentQuery();

break;

}

case 1:

{

string age1 = Convert.ToString(QueryCB.Text.Substring(0,2));

string age2 = Convert.ToString(QueryCB.Text.Substring(5,2));

dataGridView2.DataSource = controller.ageRangeQuery(age1,age2);

break;

}

case 2:

{

dataGridView2.DataSource = controller.chiefQuery(QueryTB.Text);

break;

}

case 3:

{

dataGridView2.DataSource = controller.SexQuery();

break;

}

case 4:

{

dataGridView2.DataSource = controller.SecondNameQuery();

break;

}

case 5:

{

dataGridView2.DataSource = controller.AgeQuery();

break;

}

case 6:

{

dataGridView2.DataSource = controller.initialsQuery();

break;

}

case 7:

{

dataGridView2.DataSource = controller.DepAndChiQuery();

break;

}

}

}

}

}

## 5.3 Модуль AddAndModForm

using System;

using System.Windows.Forms;

namespace Kurs\_RPK

{

public partial class AddAndModForm : Form

{

Query controller;

public AddAndModForm()

{

InitializeComponent();

controller = new Query(ConnectionStr.ConnStr);

dataGridView1.Rows.Add();

dataGridView1.AllowUserToAddRows = false;

}

//Заполнение таблицы записью выбранного работника, если было выбрано изменение данных о работнике

private void Form3\_Load(object sender, EventArgs e)

{

if (this.Text == "Изменение данных о работнике")

{

AddB.Text = "Изменить";

for (int i = 0; i < 7; i++)

{

dataGridView1.Rows[0].Cells[i].Value = Program.f2.dataGridView1.Rows[Program.f2.dataGridView1.CurrentRow.Index].Cells[i + 1].Value;

}

}

}

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

{

char[] trimC = new char[] { '\_' };

if (NotEMPT())

{ if (this.Text == "Изменение данных о работнике")

{

controller.Modify(

dataGridView1.Rows[0].Cells[0].Value.ToString().Trim(trimC),

dataGridView1.Rows[0].Cells[1].Value.ToString().Trim(trimC),

dataGridView1.Rows[0].Cells[2].Value.ToString().Trim(trimC),

dataGridView1.Rows[0].Cells[3].Value.ToString(),

dataGridView1.Rows[0].Cells[4].Value.ToString(),

dataGridView1.Rows[0].Cells[5].Value.ToString(),

dataGridView1.Rows[0].Cells[6].Value.ToString().Trim(trimC));

controller.UpdateEmployee();

MessageBox.Show("Данные о работнике успешно изменены", "Изменение данных", MessageBoxButtons.OK, MessageBoxIcon.Information);

Program.f2.Enabled = true;

Program.f2.dataGridView1.DataSource = controller.UpdateEmployee();

this.Hide();

}

else

{

controller.Add(

dataGridView1.Rows[0].Cells[0].Value.ToString().Trim(trimC),

dataGridView1.Rows[0].Cells[1].Value.ToString().Trim(trimC),

dataGridView1.Rows[0].Cells[2].Value.ToString().Trim(trimC),

dataGridView1.Rows[0].Cells[3].Value.ToString(),

dataGridView1.Rows[0].Cells[4].Value.ToString(),

dataGridView1.Rows[0].Cells[5].Value.ToString(),

dataGridView1.Rows[0].Cells[6].Value.ToString().Replace("\_",""));

controller.UpdateEmployee();

MessageBox.Show("Работник был успешно внесен в базу данных", "Работник добавлен", MessageBoxButtons.OK, MessageBoxIcon.Information);

Program.f2.Enabled = true;

Program.f2.dataGridView1.DataSource = controller.UpdateEmployee();

this.Hide();

}

}

else MessageBox.Show("Не все ячейки заполнены","Ошибка", MessageBoxButtons.OK,MessageBoxIcon.Error);

}

//Проверка на заполненность полей

private bool NotEMPT()

{

if ((dataGridView1.Rows[0].Cells[0].Value != null) &&

(dataGridView1.Rows[0].Cells[1].Value != null) &&

(dataGridView1.Rows[0].Cells[2].Value != null) &&

(dataGridView1.Rows[0].Cells[3].Value != null) &&

(dataGridView1.Rows[0].Cells[4].Value != null) &&

(dataGridView1.Rows[0].Cells[5].Value != null) &&

(dataGridView1.Rows[0].Cells[6].Value != null))

{ return true; }

else

{ return false; }

}

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

{

Program.f2.Enabled = true;

this.Hide();

}

private void Form3\_FormClosed(object sender, FormClosedEventArgs e)

{

Program.f2.Enabled = true;

this.Hide();

}

}

}

# Используемая литература

1. Методические указания к лабораторным работам по курсу РПК. – Высоцкий Л. Г./ Калининград, 2015г. – 117с.

2. Методические указания по выполнению курсовой работы по дисциплине «Разработка программных комплексов»