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 «Справочник задолжников»**

Курсовая работа по дисциплине

“Разработка и стандартизация программных средств и информационных технологий”

Пояснительная записка

Нормоконтроллер Проект выполнил

Высоцкий Л. Г. студент гр. 18 – ВТ

\_\_\_\_\_\_\_\_

Калининград

2020

**Оглавление**

[Введение 3](#_Toc59422248)

[1. Спецификация программной системы 4](#_Toc59422249)

[1.1. Дерево функций проекта 4](#_Toc59422250)

[1.2. Дерево модулей проекта 5](#_Toc59422251)

[1.3. Распределение функций системы по модулям 5](#_Toc59422252)

[1.4. Структуры используемых данных 6](#_Toc59422253)

[1.5. Видеоформы пользовательского интерфейса 6](#_Toc59422254)

[2. Руководство системного программиста 8](#_Toc59422255)

[2.1 Сведения о системе. 8](#_Toc59422256)

[2.2. Установка программы. 8](#_Toc59422257)

[2.3. Удаление программы. 8](#_Toc59422258)

[3. Руководство пользователя 9](#_Toc59422259)

[3.1. Общие сведения о программе 9](#_Toc59422260)

[3.2. Установка 9](#_Toc59422261)

[3.3.Запуск 9](#_Toc59422262)

[3.4. Инструкция по работе 9](#_Toc59422263)

[4. Справочная система 10](#_Toc59422264)

[5. Листинги модулей 14](#_Toc59422265)

[5.1 Модуль Авторизации 14](#_Toc59422266)

[5.2 Модуль Главной формы 15](#_Toc59422267)

[5.3 Модуль добавления задолженности 17](#_Toc59422268)

[Используемая литература 22](#_Toc59422269)

# Введение

Представленная в данной работе программная система «Справочник задолжников» является реализацией задания на выполнение курсовой работы по дисциплине «Разработка программных комплексов», задание сформулировано следующим образом: Справочник задолжников. Указываются номера зачетных книжек, ФИО студентов, группы, вид задолженности, дисциплина, семестр, ФИО преподавателя по дисциплине, его кафедра, зав.кафедрой, его телефон. Запросы долгов по отдельному студенту, группе, дисциплине. Два уровня доступа: администратор с паролем и пользователь..

# 1. Спецификация программной системы

## 1.1. Дерево функций проекта

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

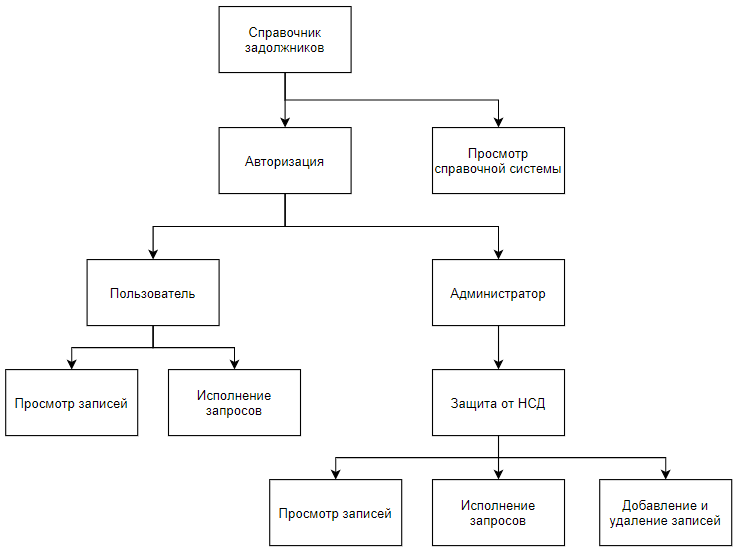


Рис. 1 Дерево функций проекта

## 1.2. Дерево модулей проекта

На рис. 2 представлено дерево модулей данной системы.

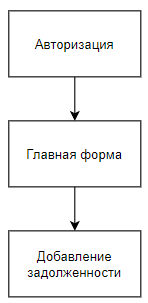


Рис. 2 Дерево модулей

## 1.3. Распределение функций системы по модулям

**Модуль Авторизации**

Включает в себя выбор режима авторизации, защиту от несанкционированного доступа, вывод информации об авторе

**Модуль Главной формы**

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

**Модуль Добавления задолженности**

Форма для добавления задолженности

## 1.4. Структуры используемых данных

|  |  |  |
| --- | --- | --- |
| Данные | Тип | Применение |
| ConnStr | string | Переменная, котороя определяет строку подключения к базе данных |
| IsAdm | integer | Переменная, которая определяет является ли пользователь администратором |
| 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 |

Структура данных в таблице «Задолженности»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Применение |
| ID | Короткий текст | Номер зачетной книжки студента |
| StudName | Короткий текст | ФИО студента |
| Group | Короткий текст | Номер группы студента |
| Type | Короткий текст | Вид задолженности |
| Semester | Короткий текст | Семестр |
| Class | Короткий текст | Дисциплина |
| TeachersName | Короткий текст | ФИО преподавателя |
| SubFaculty | Короткий текст | Кафедра преподавателя |
| Head | Короткий текст | ФИО зав. кафедрой |
| Number | Короткий текст | Номер телефона зав. кафедрой |

Структура данных в таблице «Задолженности»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Применение |
| ID | Короткий текст | Номер зачетной книжки студента |
| StudName | Короткий текст | ФИО студента |
| Group | Короткий текст | Номер группы студента |

## 1.5. Видеоформы пользовательского интерфейса

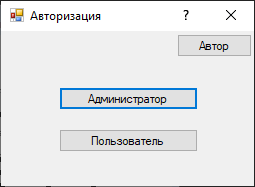


Рис.3 Видеоформа окна Авторизации

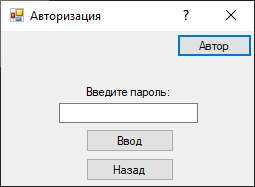


Рис.4 Видеоформа защиты от НСД при авторизации администратора

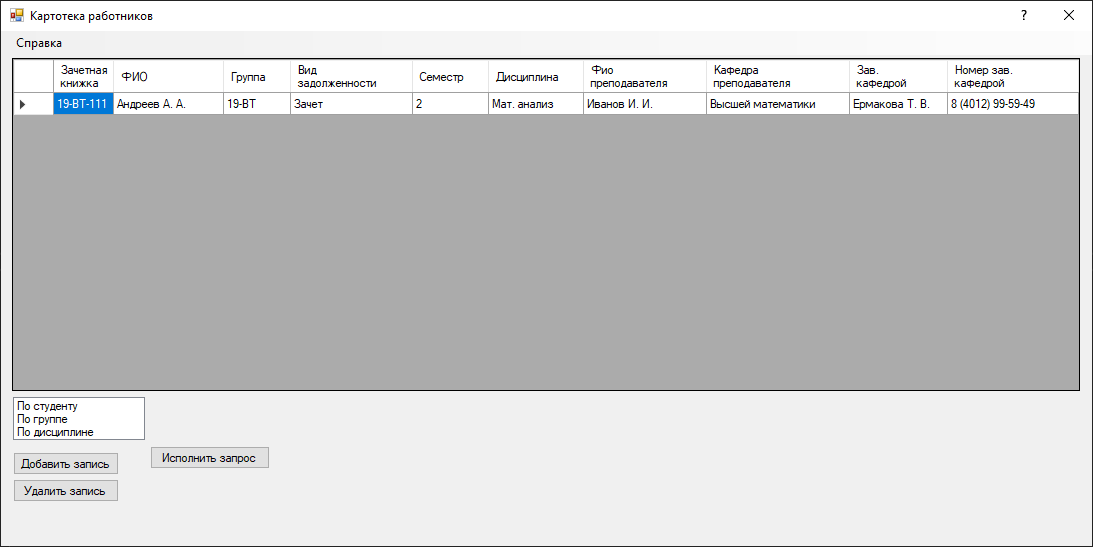


Рис. 5 Главная форма при авторизации в режиме администратора

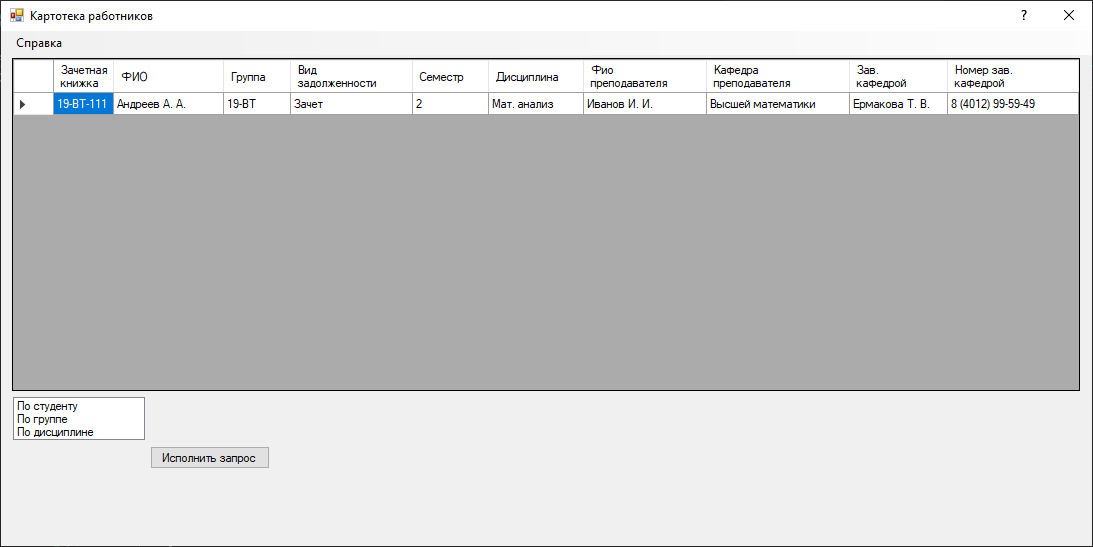


Рис.6 Главная форма при авторизации в режиме пользователя

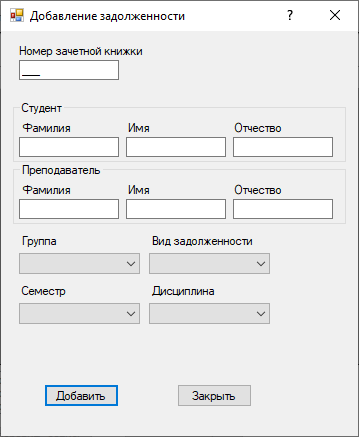


Рис.7 Видеоформа окна добавления задолженности

# 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 “Справочник задолжников.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 «Справочник задолжников» предназначена для работы в двух режимах : Пользователь и Администратор.

«Пользователь» может просматривать картотеку и исполнять запросы

«Администратор» может добавлятьи удалять задолженности.

## 3.2. Установка

Для установки ПС необходимо разархивировать архив “ Справочник задолжников.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 задолженностях студентов. Также пользователь может исполнять запросы базы данных.

Работа в режиме Администратор заключается в редактировании базы данных c информацией о задолженностях. Администратор может добавлять и удалять задолженнности.

# 4. Справочная система

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

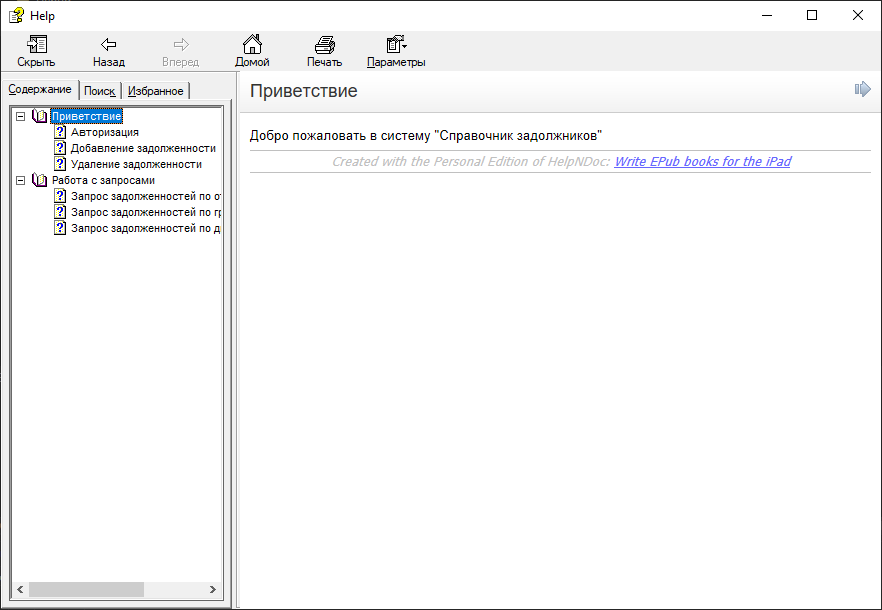


Рис. 8 Начальная страница справочной системы

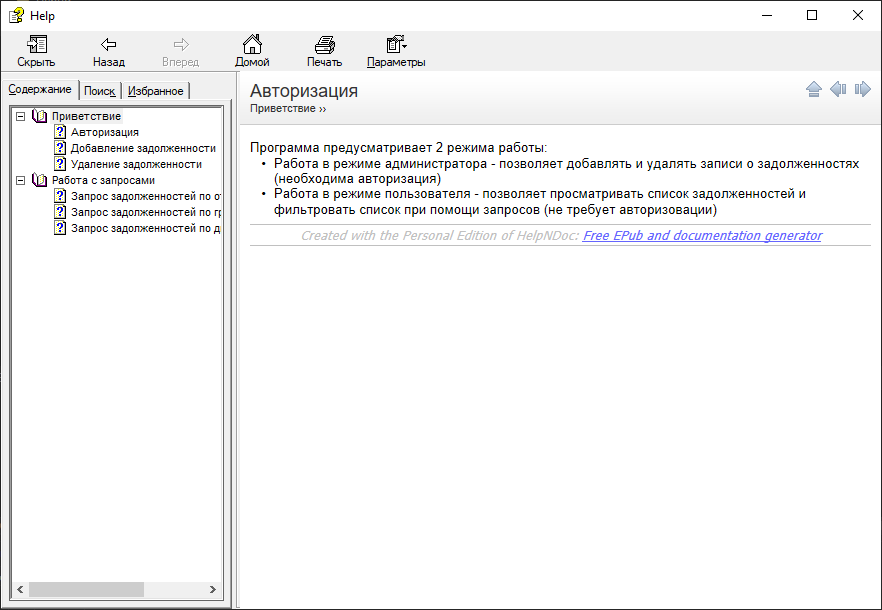


Рис. 9 Информация об авторизации

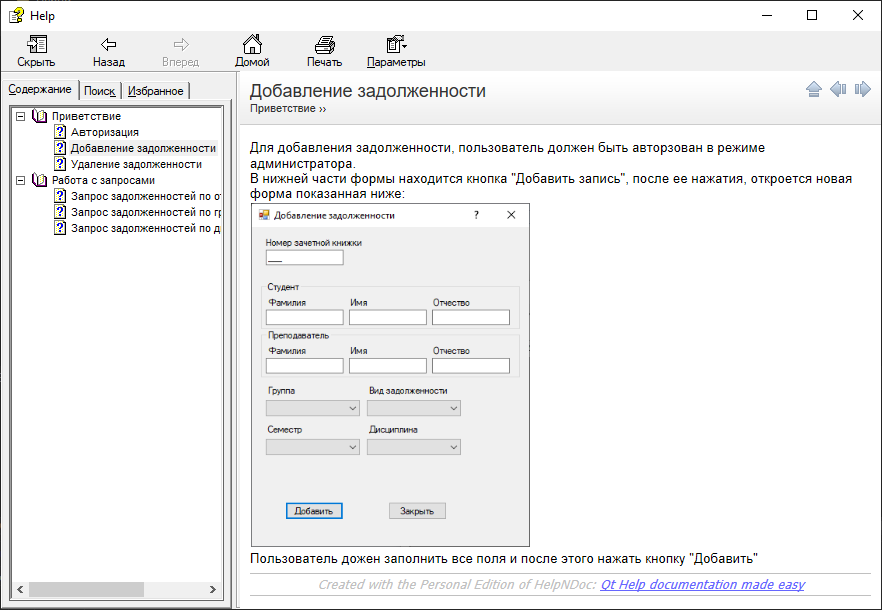


Рис. 10 Добавление задолженности

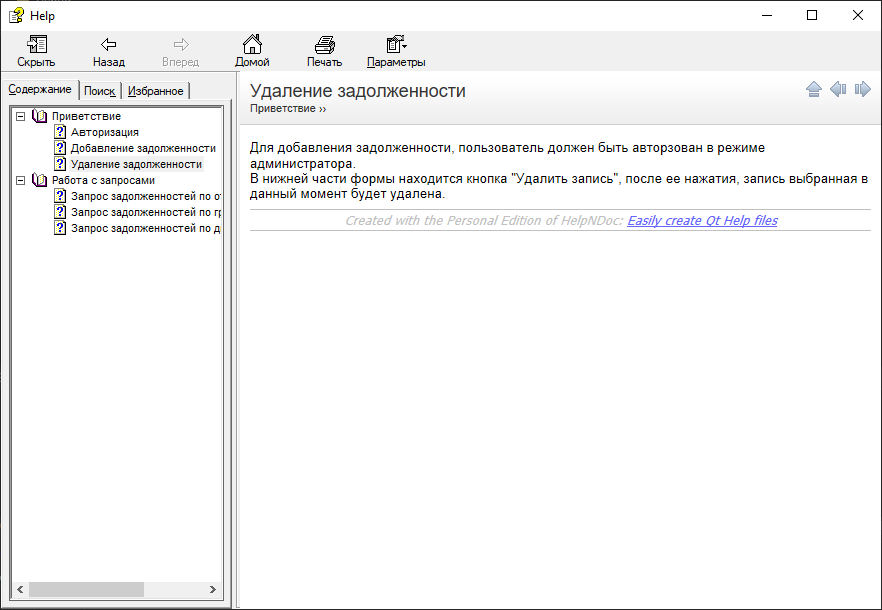
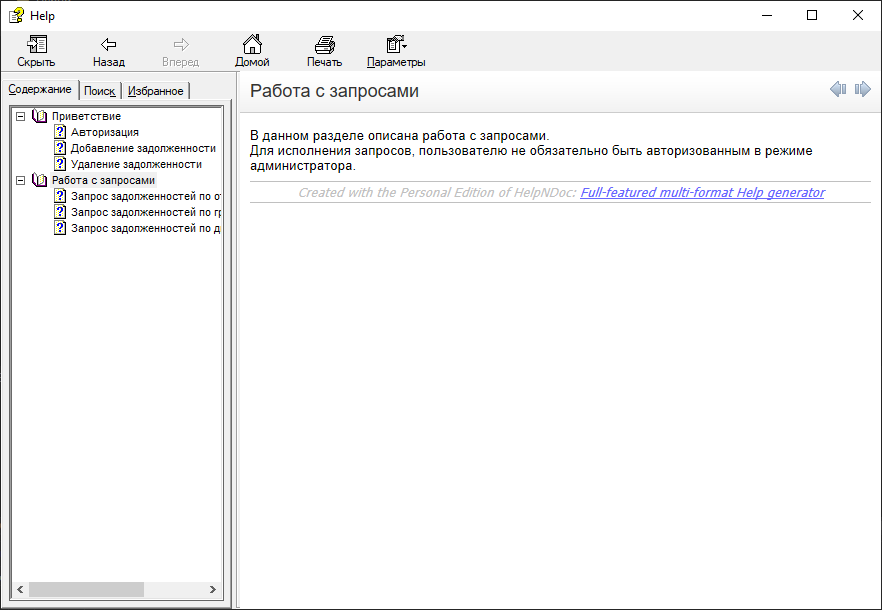


Рис. 11 Удаление задолженности

Рис. 12 Работа с запросами

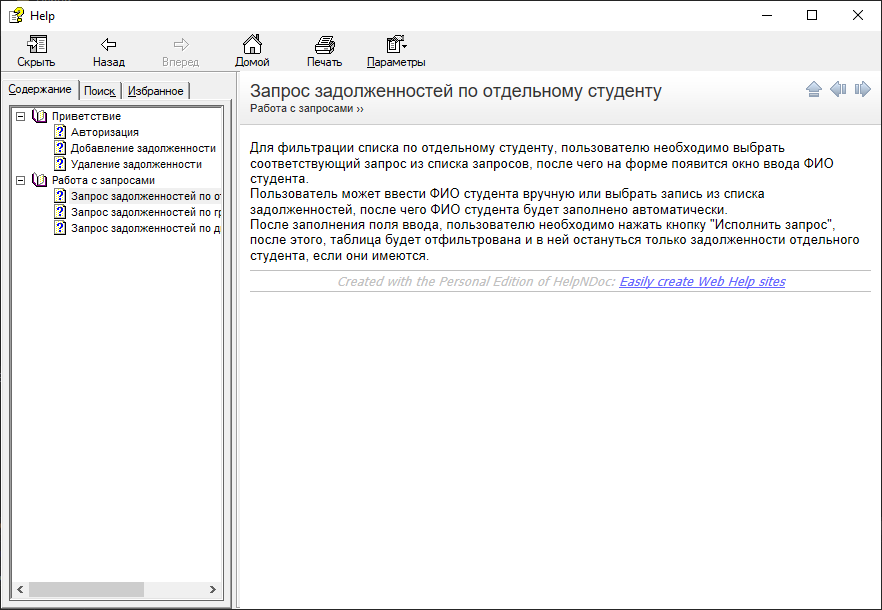


Рис. 12 Запрос задолженностей по отдельному студенту

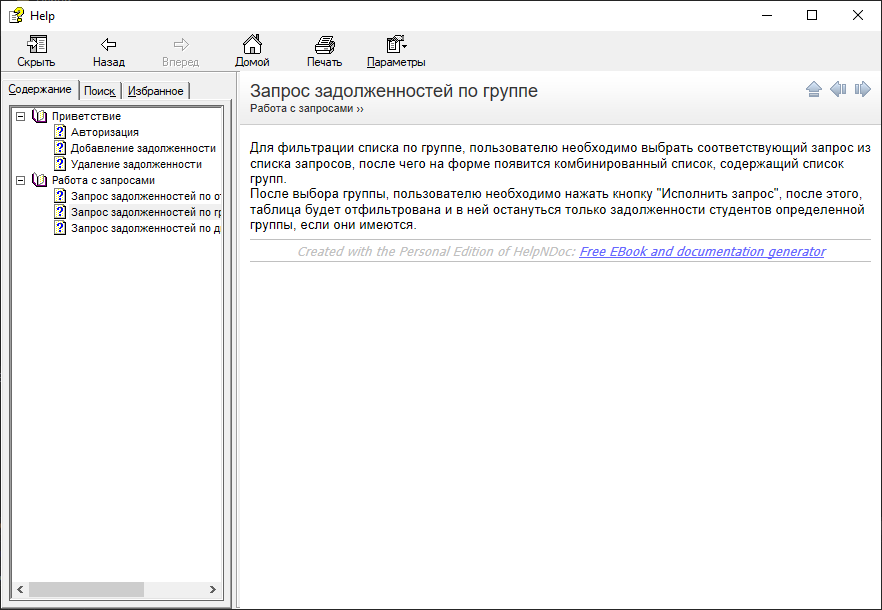


Рис. 13 Запрос задолженностей по группе

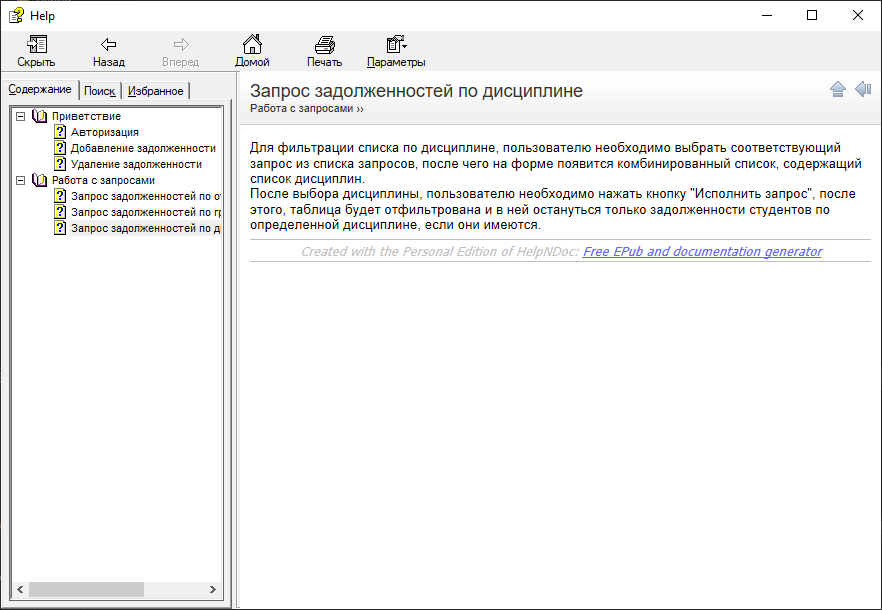


Рис. 14 Запрос задолженностей по дисциплине

# 5. Листинги модулей

## 5.1 Модуль Авторизации

using System;

using System.Windows.Forms;

namespace Kurs\_RPK

{

public partial class LogInForm : Form

{

readonly string AdmPass = "12345678";

public bool isAdm = false;

public LogInForm()

{

InitializeComponent();

Program.f1 = this;

}

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)

{

Program.f2 = new MForm();

Program.f2.Show();

this.Hide();

}

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

{

if (isAdm == true)

{

if (PasswordTB.Text == AdmPass)

{

Program.f2 = new MForm();

Program.f2.Show();

this.Hide();

}

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

}

}

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;

BackB.Visible = false;

}

}

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

{

MessageBox.Show("Система разработана студентом 18-ВТ\nАлтуховым Владиславом", "Об авторе", MessageBoxButtons.OK);

}

private void About\_HelpRequested(object sender, HelpEventArgs hlpevent)

{

Help.ShowHelp(this, "Help.chm", HelpNavigator.Topic, "newtopic1.htm");

}

}

}

## 5.2 Модуль Главной формы

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace Kurs\_RPK {

public partial class MForm : Form

{

Query controller;

public MForm()

{

InitializeComponent();

if (Program.f1.isAdm)

{

this.AddEmployee.Visible = true;

this.DeleteEmployee.Visible = true;

}

else

{

this.AddEmployee.Visible = false;

this.DeleteEmployee.Visible = false;

}

controller = new Query(ConnectionStr.ConnStr);

dataGridView1.DataSource = controller.UpdateEmployee();

}

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

{

AddAndModForm add =new AddAndModForm();

add.Text = "Добавление задолженности";

add.Show();

this.Enabled=false;

}

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

{

controller.Delete(dataGridView1.CurrentRow.Cells[0].Value.ToString(),

dataGridView1.CurrentRow.Cells[1].Value.ToString(),

dataGridView1.CurrentRow.Cells[2].Value.ToString(),

dataGridView1.CurrentRow.Cells[3].Value.ToString(),

dataGridView1.CurrentRow.Cells[4].Value.ToString(),

dataGridView1.CurrentRow.Cells[5].Value.ToString(),

dataGridView1.CurrentRow.Cells[6].Value.ToString());

MessageBox.Show("Задолженность была успешно удалена из базы данных","Удаление задолженности",MessageBoxButtons.OK,MessageBoxIcon.Information);

dataGridView1.DataSource = controller.UpdateEmployee();

}

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

{

Application.Exit();

}

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

{

switch (QueryList.SelectedIndex)

{

case 0:

{

QueryTBL.Visible = true;

QueryTBL.Text = "Введите ФИО студента или выберите запись из таблицы";

QueryMTB.Visible = false;

QueryCB.Visible = false;

QueryCBL.Visible = false;

QueryTB.Visible = true;

break;

}

case 1:

{

QueryCBL.Visible = true;

QueryCBL.Text = "Выберите номер группы";

QueryMTB.Visible = false;

QueryTB.Visible = false;

QueryTBL.Visible = false;

QueryCB.Visible = true;

QueryCB.Items.Clear();

QueryCB.Items.AddRange(new object[] {

"20-ВТ",

"20-ИЭ",

"19-ВТ",

"19-ИЭ",

"18-ВТ",

"18-ИЭ",

"17-ВТ",

"17-ИЭ" });

break;

}

case 2:

{

List<string> Classes = new List<string> {"Программирование",

"Дискретная математика",

"МНИ",

"ОС",

"РПК",

"Схемотехника",

"СИТиП",

"ИСОУ",

"ТО АСОИУ",

"ПО АСОИУ",

"Социология",

"Трудовое право",

"Правоведение",

"История",

"История региона",

"Экономика на предприятии",

"Экономика",

"Физика",

"Мат. анализ",

"Философия",

"Проф. этика",

"Экология" };

Classes.Sort();

QueryCBL.Visible = true;

QueryCBL.Text = "Выберите дисциплину";

QueryMTB.Visible = false;

QueryTB.Visible = false;

QueryTBL.Visible = false;

QueryCB.Visible = true;

QueryCB.Items.Clear();

foreach (var item in Classes)

{

QueryCB.Items.Add(item);

}

break;

}

}

}

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

{

switch (QueryList.SelectedIndex)

{

case 0:

{

dataGridView1.DataSource = controller.byStudName(QueryTB.Text);

break;

}

case 1:

{

dataGridView1.DataSource = controller.byGroup();

break;

}

case 2:

{

dataGridView1.DataSource = controller.byClass();

break;

}

}

}

private void dataGridView1\_SelectionChanged(object sender, EventArgs e)

{

if(dataGridView1.Rows.Count>0) QueryTB.Text = dataGridView1.Rows[dataGridView1.CurrentCell.RowIndex].Cells[1].Value.ToString();

}

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

{

Help.ShowHelp(this, "Help.chm", HelpNavigator.Topic, "newtopic.htm");

}

private void QueryList\_HelpRequested(object sender, HelpEventArgs hlpevent)

{

Help.ShowHelp(this, "Help.chm", HelpNavigator.Topic, "newtopic4.htm");

}

private void AddEmployee\_HelpRequested(object sender, HelpEventArgs hlpevent)

{

Help.ShowHelp(this, "Help.chm", HelpNavigator.Topic, "newtopic2.htm");

}

private void DeleteEmployee\_HelpRequested(object sender, HelpEventArgs hlpevent)

{

Help.ShowHelp(this, "Help.chm", HelpNavigator.Topic, "newtopic3.htm");

}

}

}

## 5.3 Модуль добавления задолженности

using System;

using System.Linq;

using System.Windows.Forms;

namespace Kurs\_RPK

{

public partial class AddAndModForm : Form

{

Query controller;

string[] SubF = new string[3];

string[] FAPU = new string[10] { "Программирование", "Дискретная математика", "МНИ", "ОС", "РПК", "Схемотехника", "СИТиП", "ИСОУ", "ТО АСОИУ", "ПО АСОИУ" };

string[] SOC = new string[] { "Социология", "Трудовое право", "Правоведение" };

string[] his = new string[2] { "История", "История региона" };

string[] econ = new string[2] { "Экономика на предприятии","Экономика" };

string phis = "Физика";

string math = "Мат. анализ";

string[] phil = new string[2] { "Философия", "Проф. этика" };

string eco = "Экология";

public AddAndModForm()

{

InitializeComponent();

controller = new Query(ConnectionStr.ConnStr);

}

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

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

{

}

private void subFac()

{

foreach(string s in FAPU)

{

if (Class.SelectedItem.ToString() == s)

{

SubF[0] = "СУиВТ";

SubF[1] = "Петрикин В. А.";

SubF[2] = "8 (4012) 99-59-81";

}

}

foreach (string s in SOC)

{

if (Class.SelectedItem.ToString() == s)

{

SubF[0] = "Социальных наук";

SubF[1] = "Бугакова Н. Ю.";

SubF[2] = "8 (4012) 99-53-58";

}

}

foreach (string s in his)

{

if (Class.SelectedItem.ToString() == s)

{

SubF[0] = "Истории";

SubF[1] = "Хитров А. А.";

SubF[2] = "8 (4012) 99-53-64";

}

}

foreach (string s in econ)

{

if (Class.SelectedItem.ToString() == s)

{

SubF[0] = "Экономики";

SubF[1] = "Сергеев Л. И.";

SubF[2] = "8 (4012) 69-01-20";

}

}

if (Class.SelectedItem.ToString() == phis)

{

SubF[0] = "Физики";

SubF[1] = "Синявский Н. Я.";

SubF[2] = "8 (4012) 92-51-16";

}

if (Class.SelectedItem.ToString() == math)

{

SubF[0] = "Высшей математики";

SubF[1] = "Ермакова Т. В.";

SubF[2] = "8 (4012) 99-59-49";

}

foreach (string s in phil)

{

if (Class.SelectedItem.ToString() == s)

{

SubF[0] = "Философии";

SubF[1] = "Яшина С. Л.";

SubF[2] = "8 (4012) 99-53-57";

}

}

if (Class.SelectedItem.ToString() == eco)

{

SubF[0] = "Экологии";

SubF[1] = "Шибаев С. В.";

SubF[2] = "8 (4012) 99-59-32";

}

}

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

{

string sFio = "";

string tFio = "";

if (NotEMPT())

{

sFio = sf.Text[0].ToString().ToUpper() + sf.Text.Substring(1) + " " + sn.Text.ToUpper()[0] + ". " + so.Text.ToUpper()[0] + ".";

int count = controller.newUser(Group.Text + "-" + ID.Text.Replace(" ", ""), sFio, Group.Text);

tFio = tf.Text[0].ToString().ToUpper() + tf.Text.Substring(1) + " " + tn.Text.ToUpper()[0] + ". " + to.Text.ToUpper()[0] + ".";

subFac();

if (count == 1)

{

controller.addStud(ID.Text.Replace(" ", ""), sFio, Group.Text);

controller.Add(ID.Text.Replace(" ", ""), sFio, Group.Text, Type.Text, Semester.Text, Class.Text, tFio, SubF[0], SubF[1], SubF[2]);

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

return;

}

else if (count == 2)

{

if (controller.exists(Group.Text + "-" + ID.Text.Replace(" ", ""), sFio, Group.Text, Type.Text, Semester.Text, Class.Text, tFio))

{

controller.Add(ID.Text.Replace(" ", ""), sFio, Group.Text, Type.Text, Semester.Text, Class.Text, tFio, SubF[0], SubF[1], SubF[2]);

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

}

else MessageBox.Show("Данная задолженность уже существует");

return;

}

else

{

MessageBox.Show("Номер зачетной книжки не совпадает со студентом");

return;

}

}

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

}

//Проверка на заполненность полей

private bool NotEMPT()

{

foreach(var TB in this.Controls.OfType<TextBox>())

{

if (TB.Text.Length == 0)

{

return false;

}

}

foreach (var CB in this.Controls.OfType<ComboBox>())

{

if (CB.SelectedIndex == -1)

{

return false;

}

}

return true;

}

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();

}

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

{

if(Group.SelectedIndex >= 0 && Group.SelectedIndex <= 1)

{

Semester.Items.Clear();

Semester.Items.AddRange(new object[] {"1","2" });

}

else if (Group.SelectedIndex >= 2 && Group.SelectedIndex <= 3)

{

Semester.Items.Clear();

Semester.Items.AddRange(new object[] { "1", "2","3","4" });

}

else if (Group.SelectedIndex >= 4 && Group.SelectedIndex <= 5)

{

Semester.Items.Clear();

Semester.Items.AddRange(new object[] { "3", "4", "5", "6" });

}

else if (Group.SelectedIndex >= 6 && Group.SelectedIndex <= 7)

{

Semester.Items.Clear();

Semester.Items.AddRange(new object[] { "5", "6", "7", "8" });

}

}

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

{

switch (Semester.SelectedItem.ToString())

{

case "1":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] {"Программирование","Социология","История" });

break;

}

case "2":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "Программирование", "Мат. анализ", "История региона" });

break;

}

case "3":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "Дискретная математика", "Философия", "Физика" });

break;

}

case "4":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "МНИ", "ОС", "Проф. этика" });

break;

}

case "5":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "РПК", "Экономика", "Экология" });

break;

}

case "6":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "Схемотехника", "Экономика на предприятии", "Правоведение" });

break;

}

case "7":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "СИТиП", "Социология", "Трудовое право" });

break;

}

case "8":

{

Class.Items.Clear();

Class.Items.AddRange(new object[] { "ИСОУ", "ТО АСОИУ", "ПО АСОИУ" });

break;

}

}

}

private void sf\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a' && e.KeyChar <= 'z') || e.KeyChar == ' '|| char.IsDigit(e.KeyChar)||char.IsPunctuation(e.KeyChar))

e.Handled = true;

}

}

}

# 

# 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»