**РУКОВОДСТВО ПРОГРАММИСТА**

**Приложение**

**«Информационная система университета»**

**Листов 10**

2022

**АННОТАЦИЯ**

В данном программном документе приведено руководство программиста по работе с приложением «Информационная система университета».

В разделе «Назначение и условия применения программы» указаны назначение и функции, выполняемые приложением, условия, необходимые для его работы.

В разделе «Характеристика приложения» приведено описание основных

характеристик и особенностей.

1. Назначение и условия применения программы

В последнее время все организации стараются как можно больше автоматизировать все возможные функции. Для это и разрабатываются различные информационные системы

Информационные системы имеют высокий спрос потому что организациям просто необходимо работать с огромным количеством информации, а делать это все на бумагах уже просто не представляется возможным. Информационные системы позволяют хранить, обрабатывать и использовать информацию очень легко, быстро и надёжно. Риск утратить какую-либо информацию из-за отсутствия бумаг сильно снижается

Главной задачей, которую необходимо было выполнить, было создание приложения «Информационная система университета».

Создание лёгкой по системным требованиям системы позволило бы быстро установить её на ПК, и начать использовать. Приложение способно хранить и обрабатывать всю информацию по студентам, преподавателям и успеваемости.

В ходе производственной практики было создано приложение «Информационная система университета», предназначенное для управления базой данных.

1. Характеристики и работа программы

Требования к приложению, которые были выполнены в ходе разработки:

1. Запуск возможен с любой «Windows» от 2007 года выпуска.
2. Лёгкое взаимодействие с приложением, понятное любому пользователю.
3. Надёжность хранения данных.

Требования к логике программы, которые были выполнены в ходе разработки:

1. Вход в приложение ограничен, чтобы кто угодно не мог получить доступ к информации.
2. Интуитивное главное меню со всеми необходимыми пунктами
3. Возможность удалять/изменять/добавлять/искать всю необходимую информацию
4. Возможность хранить информацию в базе данных и в любой момент получать ее.

Для того, чтобы собрать приложение требуется:

1. Установить Visual Studio, обязательно установить на нее фреймворк Windows Forms.
2. Перейти по ссылке на репозиторий GitHub  
    <https://github.com/Leneiner/IS-univer> , открыть проект через Visual Studio.
3. Подключиться к любому серверу, либо скачать и установить локальный.
4. Загрузить на сервер структуру базы данных из репозитория на GitHub.

На рисунке 1 представлена файловая архитектура приложения.

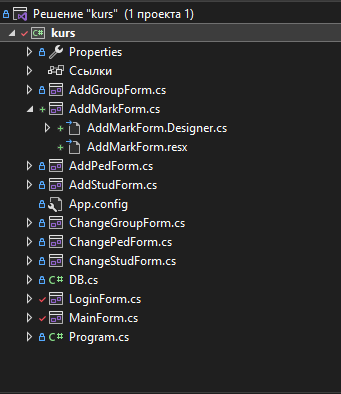


Рисунок 1 – Файловая архитектура приложения

1. Для работы необходимы все файлы, представленные на рисунке 1/

При создании проекта использовался язык C#.

После авторизации становится доступен основной функционал приложения – меню со всем функционалом. На рисунке 2 представлен код проверки и авторизации пользователя.

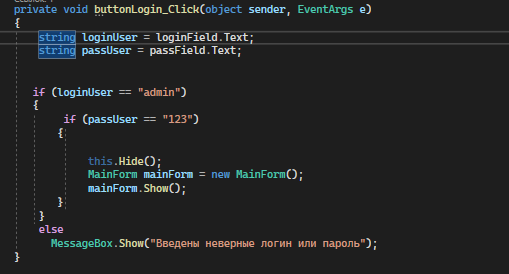


Рисунок 2 – Код проверки и авторизации пользователя.

При создании использовался фреймворк Windows Forms. Основная форма где происходит управление данными – MainForm. В Main.cs описан вывод данных их базы данных, описаны кнопки поиска и сортировки любой информации а также кнпоки перехода на формы позволяющие добавлять или изменять информацию.

На рисунке 3 представлена реализация добавления данных в базу данных.

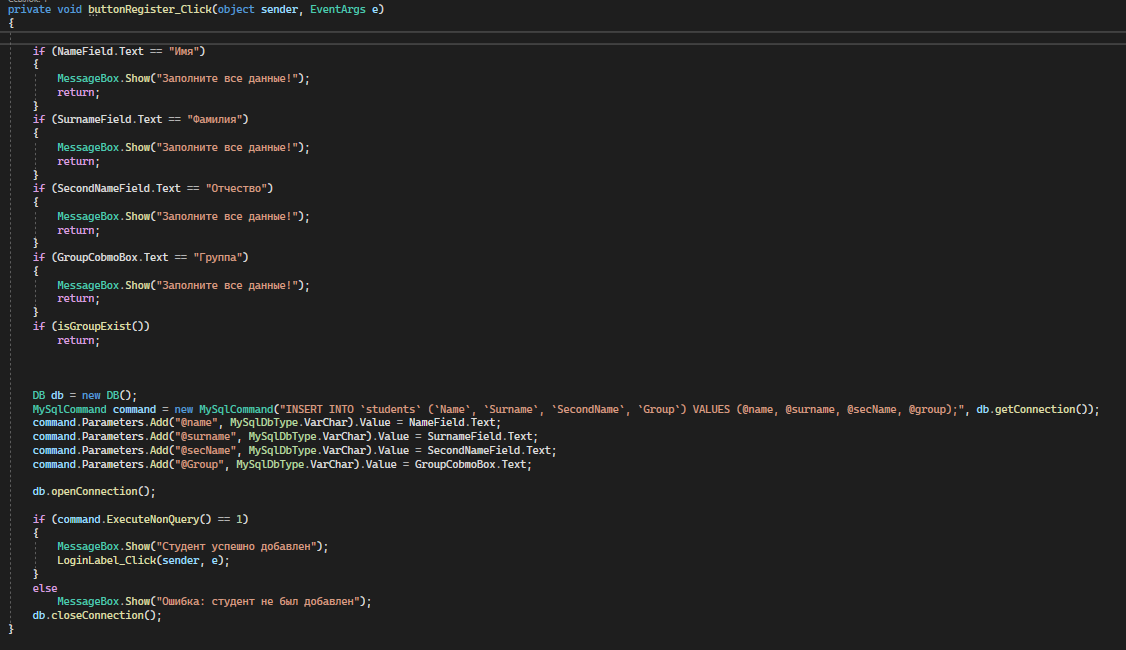


Рисунок 3 – Реализация просчёта соприкосновения

На рисунке видно, что метод проверяет наличие введеных данных и отправляет базе данных SQL запрос на добавление новой записи.

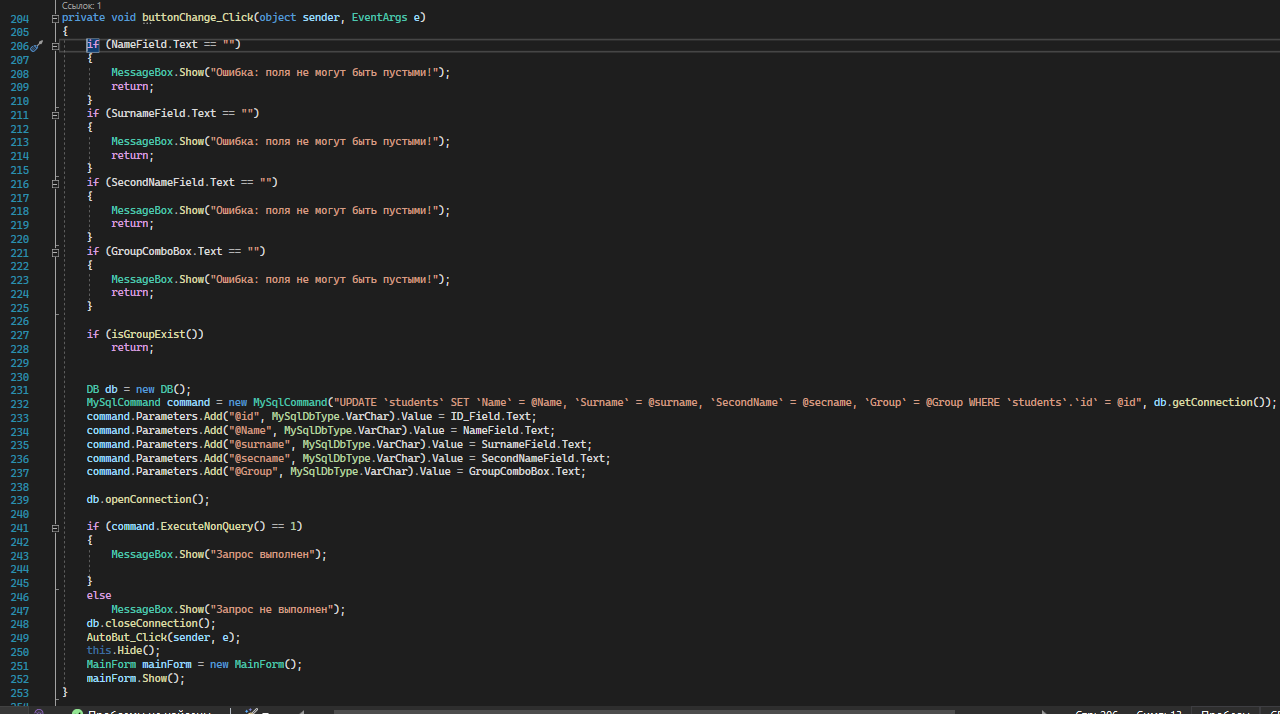


Рисунок 4 – Изменение данных базы данных

На данном рисунке также видно, про что происходит проверка корректности введеных данных и отправка SQL запроса базе данных на изменение данных. Если база возвращает положительный ответ это транслируется пользователю.

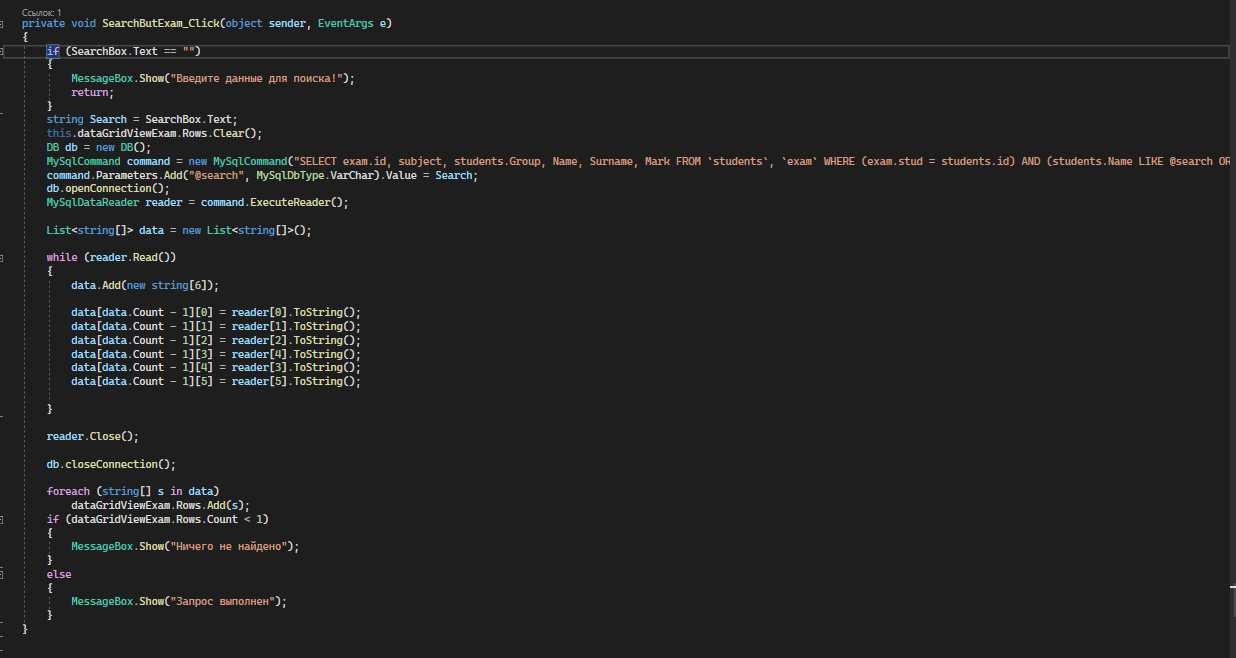


Рисунок 5 – Реализация поиска данных

На рисунке 5 представлен код отправки запроса выборки данных из базы данных. Информацию из базы можно выбрать по любому признаку, запрос сравнивает каждый атрибут всех записей с ключевым словом и если таковое найдено, заполняет элемент DataGridView информацией.

ЛИСТИНГ ПРОГРАММЫ

**«Метод Delete»**

private void DelMarkBut2\_Click(object sender, EventArgs e)

{

string ID = DelMarkField.Text;

DB db = new DB();

MySqlCommand command = new MySqlCommand("DELETE FROM `exam` WHERE `exam`.`id` = @idExam", db.getConnection());

command.Parameters.Add("@idExam", MySqlDbType.VarChar).Value = ID;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show("Оценка успешно удален");

DelMarkPanel.Enabled = false;

DelMarkPanel.Visible = false;

button1\_Click(sender, e);

}

else

MessageBox.Show("Такого ID не существует");

db.closeConnection();

}

**«Подключение к базе данных»**

internal class DB

{

MySqlConnection connection = new MySqlConnection ("server=localhost;port=3306;username=root;password=root;database=kurs");

public void openConnection()

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

}

public void closeConnection()

{

if (connection.State == System.Data.ConnectionState.Open)

connection.Close();

}

public MySqlConnection getConnection()

{

return connection;

}

}