**ГАПОУ РК «Петрозаводский техникум городского хозяйства»**

КУРСОВОЙ ПРОЕКТ

по МДК 01.01

на тему: «Разработка БД и приложения деятельности студенческого общежития)»

Выполнил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

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

*специальность, курс, № группы*

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

Дата сдачи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата проверки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

*подпись руководителя*

Петрозаводск, 2021

**Содержание**

Общие сведения........................................................................................................... 3

Введение....................................................................................................................... 4

Анализ предметной области ...................................................................................... 5

Состав типового варианта приложения .................................................................. 6

Постановка задачи на разработку............................................................................ 7

Обоснование выбора языка программирования .................................................... 8

Построение моделей данных.................................................................................... 11

Проектирование пользовательского интерфейса................................................... 12

Заключение ................................................................................................................ 16

Список литературы и источников ........................................................................... 17

**Тема курсовой работы**

**«Разработка БД и приложения деятельности студенческого общежития»**

**Функции системы:**

1. Просматривать данные о жильцах.
2. Возможность добавлять, редактировать, удалять информацию о жильцах.
3. Сохранять информацию в базу данных.
4. Возможность просматривать информацию о определённом жильце благодаря поиску.
5. Автоматический контроль количества жильцов в комнате.

**Объект исследования:** Студенческое общежитие, программные средства для автоматизации бухгалтерского учёта данных о жильцах.

**Цель работы:** разработать информационную систему и приложение, осуществляющее автоматизацию бухгалтерского учёта данных о жильцах общежития.

**Технология разработки:** среда разработки, компилятор – Visual Studio Community 2019, Visual Studio Community 2022, MySQL Workbench 8.0. Исходным языком программирования является C#.

**Результат работы:** разработана информационная система управления студенческим общежитием.

**Предмет исследования:** документооборот студенческого общежития

**Метод исследования:** поиск, анализ, обобщение информации о студенческом общежитии; изучение и анализ документации, литературы.

**Введение**

С развитием информационных технологий компьютеры, с их расширенными функциональными возможностями, активно применяются в различных сферах человеческой деятельности, связанных с обработкой информации, представлением данных. Не стала исключением и сфера образования. Возросшие объемы обработки информации, требования к информационному взаимодействию отраслей образовательной сферы стали факторами, определяющими необходимость создания и внедрения автоматизированных информационных систем.

Для исследования выбрана организация, осуществляющая социальное обслуживание населения. Основным направлением деятельности является обеспечение студентов временем жильём.

Передо мной были поставлены следующие задачи:

1. Собрать необходимую информацию о деятельности студенческого общежития.
2. Разработать алгоритм решения поставленной цели.
3. Выбрать наиболее подходящую среду программирования.
4. Разработать пользовательский интерфейс и основные функциональные возможности программного средства.
5. Определить варианты использования разрабатываемой системы.
6. Выполнить анализ средств разработки и существующих СУБД для поставленной цели.
7. Реализовать возможность добавления, удаления, изменения информации о жильцах студенческого общежития.
8. Реализовать просмотр данных о жильцах.
9. Реализовать возможность сохранения информации в базу данных.

В результате, на основе выполненного анализа, была разработана информационная система, автоматизирующая деятельность студенческого общежития, способное сохранять информацию в базу данных, также изменять и удалять ее.

**Анализ предметной области**

**Типовой вариант приложения, автоматизирующего работу студенческого общежития, состоит из следующих частей**

**Главная форма**, содержащая основные сведения того, что есть в приложении. Данное окно предназначено для перехода к любой существующей форме в приложении.

**Форма добавления** данных о жильцах. В этом окне производится заполнение основной информации о жильце. Данные сохраняются в базу данных, которые впоследствии можно изменить.

**Форма просмотра, добавления, изменения, удаления данных о жильцах.** Данное окно предназначено для просмотра, добавления и изменения информации о жильцах.

**Постановка задач на разработку**

Целью разработки является создание приложения с базой данных, которое будет автоматизировать учёт жильцов студенческого общежития, хранить информацию о ранее проживающих жильцах, а также иметь возможность редактировать, удалять, вносить данные в базу данных.

Данные о проживающих будут вноситься в программное приложение, вся необходимая информация будет под рукой. Внесен ли уже тот или иной студент в базу данных можно проверить, осуществив поиск по его фамилии, таким образом, время поиска значительно сократится. Там же будет возможность просмотра всей информации о человеке.

Основные требования к приложению.

**-** Добавление информацию о вновь заселенных студентах.

**-** Изменение информации о выселенных студентах (заменить дату

выселения).

- Получение списка свободных мест

- Получение информации о комнате

- Получение списка проживающих

**Основание выбора языка программирования**

Для реализации приложения «Система управления общежитием» была выбрана система программирования Visual studio 2019 windows forms на языке С#, так как она представляет наиболее широкие возможности для программирования приложений ОС Windows, а также имеет интуитивный интерфейс и проста в использовании.

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Преимущества Visual Studio:

- Поддержка множества языков программирования; - Кроссплатформенность разработки;

- Встроенный контроль за выполнением многопоточного кода; - Наличие бесплатной редакции Community;

- Запись происходящего во время отладки (функция IntelliTrace); Основные возможности:

1. Редактирование, профилирование и отладка кода;

2. Нумерация строк в редакторе

3. Поддержка языков Visual Basic, C#, C++, Java Script, HTML5, Python

4. Автоматическое завершение кода и подсказки

5. Рефакторинг и реструктуризация исходных текстов программ

6. Система скачиваемых расширений для дополнительных функций

7. Встроенные тесты производительности

8. Эмуляция мобильных платформ

9. Редактируемое меню внешних инструментов.

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений. В Windows Forms можно разрабатывать графически сложные приложения, которые просто развертывать, обновлять, и с которыми удобно работать как в автономном режиме, так и в сети. Приложения Windows Forms могут получать доступ к локальному оборудованию и файловой системе компьютера, на котором работает приложение.

Для работы с данными используются системы управления базами данными. Для разработки приложения была выбрана локальная база данных в MySQL Workbench 8.0

MySQL Workbench — это унифицированный визуальный инструмент для архитекторов баз данных и разработчиков БД. MySQL Workbench предоставляет возможность моделирование данных, разработку SQL и комплексные инструменты администрирования для конфигурации сервера, администрирования пользователей, резервного копирования и многое другое. MySQL Workbench доступен на Windows, Linux и MacOSX.

Преимущества MySQL Workbench:

- MySQL Workbench позволяет DBA, разработчику или архитектору данных визуально проектировать, моделировать, генерировать и управлять базами данных. Он включает в себя все, что нужно разработчику моделей данных для создания сложных ER-моделей, прямого и обратного проектирования, а также предоставляет ключевые функции для выполнения сложных задач управления изменениями и документирования, которые обычно требуют много времени и усилий.

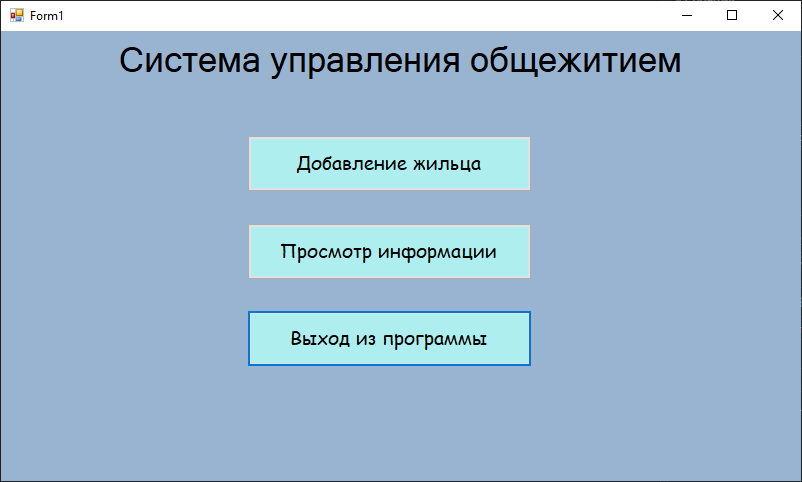
- MySQL Workbench предоставляет визуальные инструменты для создания, выполнения и оптимизации SQL-запросов. Редактор SQL предоставляет цветовую подсветку синтаксиса, автозаполнение, повторное использование фрагментов SQL и историю выполнения SQL. Панель подключения к базе данных позволяет разработчикам легко управлять стандартными подключениями к базе данных, включая MySQL Fabric. Обозреватель объектов обеспечивает мгновенный доступ к схеме базы данных и объектам.

MySQL Workbench предоставляет визуальную консоль, чтобы легко администрировать среды MySQL и получить лучшую видимость в базах данных. Разработчики и администраторы баз данных могут использовать визуальные инструменты для настройки серверов, администрирования пользователей, выполнения резервного копирования и восстановления, проверки данных аудита и просмотра работоспособности базы данных.

**Построение моделей данных**

**Проектирование пользовательского интерфейса**

Основные формы программного приложения «Система управления общежитием»:

****

**3**

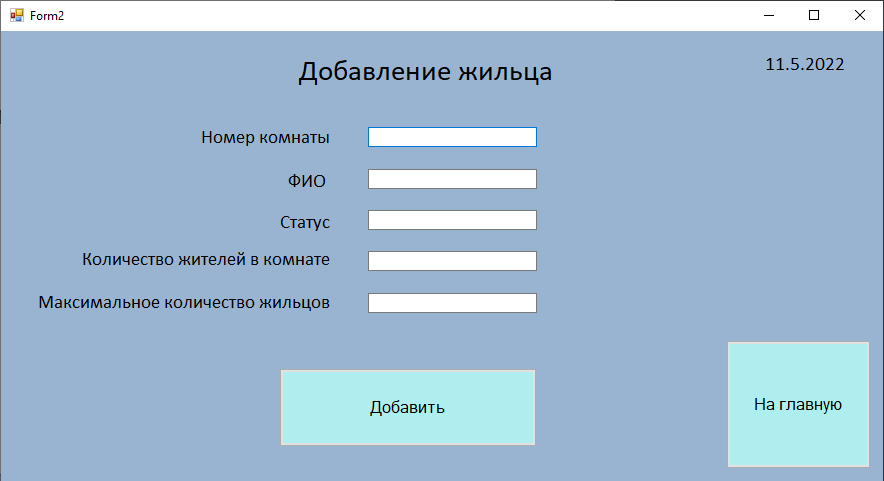
**2**

**1**

Рис.1 - Главная форма приложения

Обозначения:

1. Кнопка для перехода на форму добавления нового жильца.
2. Кнопка для перехода на форму просмотра информации о проживающих.
3. Кнопка выхода из программы.



**7**

**6**

**5**

**4**

**3**

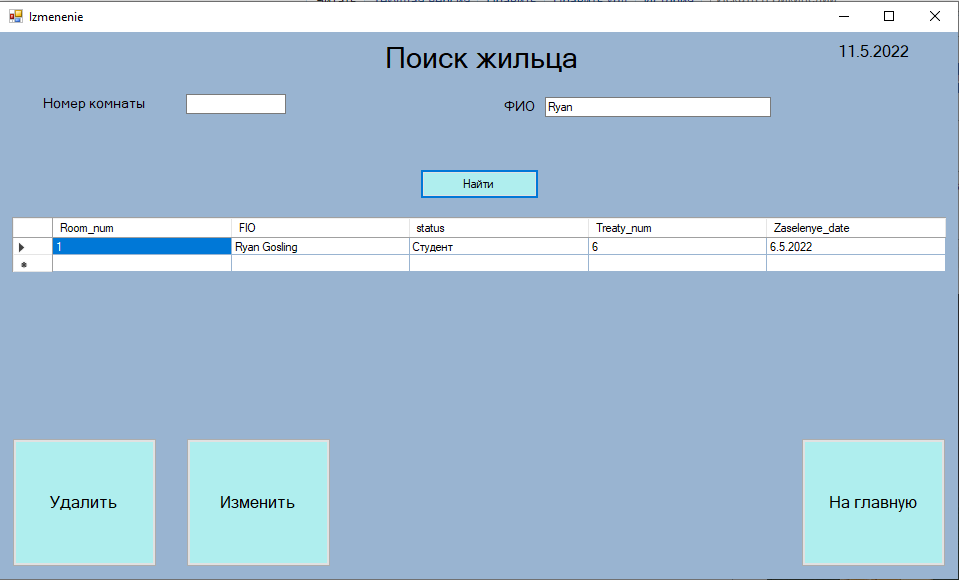
**2**

**1**

Рис.2 – Форма добавления информации о новом жильце.

Обозначения:

1. Поле для ввода комнаты заселяемого
2. Поле для ввода ФИО заселяемого
3. Поле для ввода статуса заселяемого
4. Поле отображения текущего количества жителей в комнате
5. Поле отображения максимального количества жителей в комнате
6. Кнопка добавление информации в базу данных
7. Кнопка перехода на главную форму



**7**

**6**

**5**

**4**

**3**

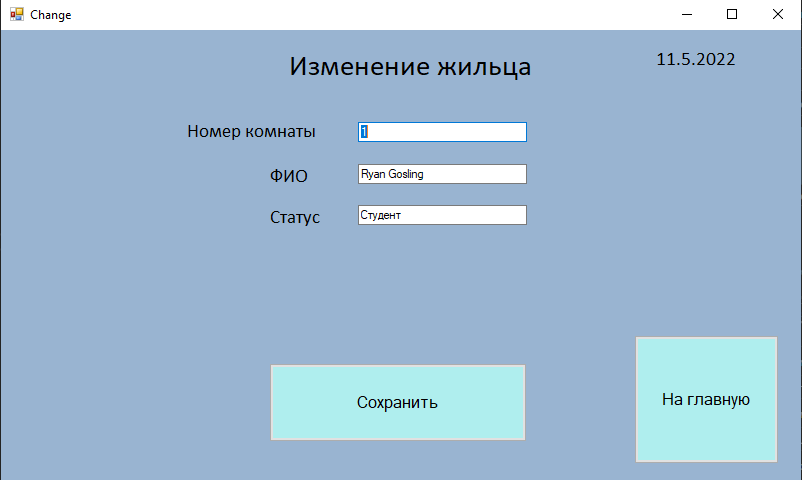
**2**

**1**

Рис.3 – Форма просмотра информации о проживающих

Обозначения:

1. Поле для ввода номера комнаты для поиска
2. Поле для ввода ФИО для поиска
3. Кнопка выполнения поиска
4. Таблица для отображения результатов поиска
5. Кнопка удаления информации из базы данных
6. Кнопка для перехода на форму изменения информации о проживающем
7. Кнопка для перехода на главную форму

****

**5**

**4**

**3**

**2**

**1**

Рис.4 – Форма изменения информации о проживающем

Обозначения:

1. Поле для изменения комнаты проживания жильца
2. Поле для изменения ФИО проживающего
3. Поле для изменения статуса проживающего
4. Кнопка для сохранения информации в базу данных
5. Кнопка возврата на главную форму

**Заключение**

В итоге было создано приложение «Система управления общежитием» в среде разработки Visual Studio 2019 – 2022 на языке программирования C# которое позволяет автоматически вести учёт проживающих в общежитии студентов, оперативно получать информацию о конкретном человеке или комнате, а также изменять информацию о уже проживающих студентах. В бедующем приложение планируется дорабатывать.

В данной работе была изучена предметная область, выполнен анализ работы студенческого общежития. Были определены функции которыми должно обладать приложение, после в связи с требованиями была выбрана среда разработки приложения и базы данных к нему.

**Список литературы**

1. Бьюли. А. «Изучаем SQL. Генерация, выборка и обработка данных» Вильямс 2021
2. Грабер М. «SQL для простых смертных» Лори 2021
3. Бхаргава А. «Грокаем Алгоритмы» Питер 2018
4. Форум <https://cyberforum.ru>
5. Форум <https://stackoverflow.co>

**Приложение 1**

Код главной формы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Kursovaya

{

public partial class Glavnaya : Form

{

public Glavnaya()

{

InitializeComponent();

}

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

{

Application.Exit();

}

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

{

Form f2 = new Izmenenie();

f2.Show();

this.Hide();

}

private void button3\_Click(object sender, EventArgs e)

{

Form f3 = new Dobavlenie();

f3.Show();

this.Hide();

}

}

}

**Приложение 2**

Код формы добавления информации:

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Kursovaya

{

public partial class Dobavlenie : Form

{

public class LibWrap

{

[DllImport("Kernel32.dll")]

public static extern void GetLocalTime([In, Out] SystemTime st);

}

[StructLayout(LayoutKind.Sequential)]

public class SystemTime

{

public ushort year;

public ushort month;

public ushort weekday;

public ushort day;

public ushort hour;

public ushort minute;

public ushort second;

public ushort millisecond;

}

public static string connStr = "server=127.0.0.1;user=root;database=obshaga;password=mysql";

public MySqlConnection conn = new MySqlConnection(connStr);

public Dobavlenie()

{

InitializeComponent();

conn.Open();

}

private void Form2\_Load(object sender, EventArgs e)

{

SystemTime st = new SystemTime();

LibWrap.GetLocalTime(st);

label1.Text = st.day.ToString() + "." + st.month.ToString() + "." + st.year.ToString();

}

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

{

String roomnum = textBox1.Text;

string query = "SELECT \* from zhitelxrooms WHERE Room\_num LIKE '"+ roomnum +"'";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

textBox5.Text = reader.GetString(1);

}

reader.Close();

string query1 = "SELECT COUNT(Room\_num) from rooms WHERE Room\_num LIKE '"+ roomnum +"'";

MySqlCommand cmd1 = new MySqlCommand(query1, conn);

cmd1.ExecuteNonQuery();

var Vil\_count = cmd1.ExecuteScalar().ToString();

textBox4.Text = System.Convert.ToString(Vil\_count);

}

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

{

String roomnum = textBox1.Text;

String FIO = textBox2.Text;

String status = textBox3.Text;

int Vil\_count = Convert.ToInt32(textBox4.Text);

String Vil\_max = textBox5.Text;

String Zaselenie\_date = label1.Text;

Vil\_count += 1;

string query = "INSERT INTO rooms ( Room\_num, FIO, Status, Vil\_count, Vil\_max, Zaselenie\_date, Vyselenie\_date) VALUES ('"+ roomnum +"', '"+ FIO +"', '"+ status +"','"+ Vil\_count +"','"+ Vil\_max +"','"+ Zaselenie\_date +"', 0 )";

string query1 = "INSERT INTO archive ( Room\_num, FIO, Status, Vil\_count, Vil\_max, Zaselenie\_date, Vyselenie\_date) VALUES ('" + roomnum + "', '" + FIO + "', '" + status + "','" + Vil\_count + "','" + Vil\_max + "','" + Zaselenie\_date + "', 0 )";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlCommand cmd1 = new MySqlCommand(query1, conn);

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

}

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

{

Form f1 = new Glavnaya();

f1.Show();

this.Close();

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox5.Text == textBox4.Text)

{

label8.Text = "В этой комнате максимальное количество жильцов!";

label8.ForeColor = Color.Red;

button1.BackColor = Color.Gray;

button1.Enabled = false;

}

else

{

button1.BackColor = Color.PaleTurquoise;

button1.Enabled = true;

label8.Text = "";

}

if ((textBox1.Text != "" && textBox5.Text != textBox4.Text))

{

label8.Text = "В этой комнате достаточно мест.";

label8.ForeColor = Color.Green;

}

}

}

}

**Приложение 3**

Код формы просмотра информации:

using System;

using System.Collections.Generic;

using System.Data.OleDb;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Runtime.InteropServices;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Kursovaya

{

public partial class Izmenenie : Form

{

public class LibWrap

{

[DllImport("Kernel32.dll")]

public static extern void GetLocalTime([In, Out] SystemTime st);

}

[StructLayout(LayoutKind.Sequential)]

public class SystemTime

{

public ushort year;

public ushort month;

public ushort weekday;

public ushort day;

public ushort hour;

public ushort minute;

public ushort second;

public ushort millisecond;

}

public static string izm\_num;

public static string connStr = "server=127.0.0.1;user=root;database=obshaga;password=mysql";

public MySqlConnection conn = new MySqlConnection(connStr);

public Izmenenie()

{

InitializeComponent();

conn.Open();

}

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

{

dataGridView1.Visible = true;

dataGridView1.Rows.Clear();

try

{

conn.Open();

}

catch

{

}

String roomnum = textBox1.Text;

String FIO = textBox2.Text;

string query = "SELECT \* FROM rooms where Room\_num LIKE \'" + roomnum + "\' ";

MySqlCommand command = new MySqlCommand(query, conn);

MySqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[8]);

data[data.Count - 1][0] = reader.GetString(0);

data[data.Count - 1][1] = reader.GetString(1);

data[data.Count - 1][2] = reader.GetString(2);

data[data.Count - 1][3] = reader.GetString(5);

data[data.Count - 1][4] = reader.GetString(6);

}

reader.Close();

foreach (string[] s in data)

dataGridView1.Rows.Add(s);

if (textBox1.Text == "")

{

reader.Close();

string queryZ = "SELECT \* FROM rooms where FIO LIKE \'%" + FIO + "%\' ";

MySqlCommand commandZ = new MySqlCommand(queryZ, conn);

MySqlDataReader readerZ = commandZ.ExecuteReader();

List<string[]> dataZ = new List<string[]>();

while (readerZ.Read())

{

dataZ.Add(new string[8]);

dataZ[dataZ.Count - 1][0] = readerZ.GetString(0);

dataZ[dataZ.Count - 1][1] = readerZ.GetString(1);

dataZ[dataZ.Count - 1][2] = readerZ.GetString(2);

dataZ[dataZ.Count - 1][3] = readerZ.GetString(5);

dataZ[dataZ.Count - 1][4] = readerZ.GetString(6);

}

readerZ.Close();

foreach (string[] s in dataZ)

dataGridView1.Rows.Add(s);

}

}

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

{

SystemTime st = new SystemTime();

LibWrap.GetLocalTime(st);

label4.Text = st.day.ToString() + "." + st.month.ToString() + "." + st.year.ToString();

}

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

{

Form f1 = new Glavnaya();

f1.Show();

this.Hide();

}

private void button3\_Click(object sender, EventArgs e)

{

string delet = dataGridView1.SelectedCells[3].Value.ToString();

string query = "DELETE FROM rooms where Treaty\_num = '" + delet + "'";

string query1 = "UPDATE archive SET Vyselenie\_date = '" + label4.Text + "' WHERE Treaty\_num = '" + delet + "'";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlCommand cmd1 = new MySqlCommand(query1, conn);

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

}

private void button4\_Click(object sender, EventArgs e)

{

var izm = dataGridView1.SelectedCells[3].Value.ToString();

izm\_num = izm;

Form fch = new Change();

fch.Show();

this.Hide();

}

}

public class RoundButton : Control

{

public Color BackColor2 { get; set; }

public Color ButtonBorderColor { get; set; }

public int ButtonRoundRadius { get; set; }

public Color ButtonHighlightColor { get; set; }

public Color ButtonHighlightColor2 { get; set; }

public Color ButtonHighlightForeColor { get; set; }

public Color ButtonPressedColor { get; set; }

public Color ButtonPressedColor2 { get; set; }

public Color ButtonPressedForeColor { get; set; }

private bool IsHighlighted;

private bool IsPressed;

public RoundButton()

{

Size = new Size(100, 40);

ButtonRoundRadius = 30;

BackColor = Color.Gainsboro;

BackColor2 = Color.Silver;

ButtonBorderColor = Color.Black;

ButtonHighlightColor = Color.Orange;

ButtonHighlightColor2 = Color.OrangeRed;

ButtonHighlightForeColor = Color.Black;

ButtonPressedColor = Color.Red;

ButtonPressedColor2 = Color.Maroon;

ButtonPressedForeColor = Color.White;

}

protected override CreateParams CreateParams

{

get

{

CreateParams createParams = base.CreateParams;

createParams.ExStyle |= 0x00000020; // WS\_EX\_TRANSPARENT

return createParams;

}

}

protected override void OnPaint(PaintEventArgs e)

{

e.Graphics.SmoothingMode = SmoothingMode.HighQuality;

var foreColor = IsPressed ? ButtonPressedForeColor : IsHighlighted ? ButtonHighlightForeColor : ForeColor;

var backColor = IsPressed ? ButtonPressedColor : IsHighlighted ? ButtonHighlightColor : BackColor;

var backColor2 = IsPressed ? ButtonPressedColor2 : IsHighlighted ? ButtonHighlightColor2 : BackColor2;

using (var pen = new Pen(ButtonBorderColor, 1))

e.Graphics.DrawPath(pen, Path);

using (var brush = new LinearGradientBrush(ClientRectangle, backColor, backColor2, LinearGradientMode.Vertical))

e.Graphics.FillPath(brush, Path);

using (var brush = new SolidBrush(foreColor))

{

var sf = new StringFormat { Alignment = StringAlignment.Center, LineAlignment = StringAlignment.Center };

var rect = ClientRectangle;

rect.Inflate(-4, -4);

e.Graphics.DrawString(Text, Font, brush, rect, sf);

}

base.OnPaint(e);

}

protected override void OnPaintBackground(PaintEventArgs e)

{

}

protected override void OnMouseEnter(EventArgs e)

{

base.OnMouseEnter(e);

IsHighlighted = true;

Parent.Invalidate(Bounds, false);

Invalidate();

}

protected override void OnMouseLeave(EventArgs e)

{

base.OnMouseLeave(e);

IsHighlighted = false;

IsPressed = false;

Parent.Invalidate(Bounds, false);

Invalidate();

}

protected override void OnMouseDown(MouseEventArgs e)

{

base.OnMouseDown(e);

Parent.Invalidate(Bounds, false);

Invalidate();

IsPressed = true;

}

protected override void OnMouseUp(MouseEventArgs e)

{

base.OnMouseUp(e);

Parent.Invalidate(Bounds, false);

Invalidate();

IsPressed = false;

}

protected GraphicsPath Path

{

get

{

var rect = ClientRectangle;

rect.Inflate(-1, -1);

return GetRoundedRectangle(rect, ButtonRoundRadius);

}

}

public static GraphicsPath GetRoundedRectangle(Rectangle rect, int d)

{

var gp = new GraphicsPath();

gp.AddArc(rect.X, rect.Y, d, d, 180, 90);

gp.AddArc(rect.X + rect.Width - d, rect.Y, d, d, 270, 90);

gp.AddArc(rect.X + rect.Width - d, rect.Y + rect.Height - d, d, d, 0, 90);

gp.AddArc(rect.X, rect.Y + rect.Height - d, d, d, 90, 90);

gp.CloseFigure();

return gp;

}

}

}

**Приложение 4**

Код формы изменения информации:

using MySqlConnector;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Kursovaya

{

public partial class Change : Form

{

public class LibWrap

{

[DllImport("Kernel32.dll")]

public static extern void GetLocalTime([In, Out] SystemTime st);

}

[StructLayout(LayoutKind.Sequential)]

public class SystemTime

{

public ushort year;

public ushort month;

public ushort weekday;

public ushort day;

public ushort hour;

public ushort minute;

public ushort second;

public ushort millisecond;

}

public static string connStr = "server=127.0.0.1;user=root;database=obshaga;password=mysql";

public MySqlConnection conn = new MySqlConnection(connStr);

public Change()

{

InitializeComponent();

conn.Open();

}

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

{

SystemTime st = new SystemTime();

LibWrap.GetLocalTime(st);

label1.Text = st.day.ToString() + "." + st.month.ToString() + "." + st.year.ToString();

string query = "SELECT \* FROM rooms WHERE Treaty\_num LIKE '" + Izmenenie.izm\_num + "'";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

textBox1.Text = reader.GetString(0);

textBox2.Text = reader.GetString(1);

textBox3.Text = reader.GetString(2);

}

reader.Close();

}

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

{

Form f1 = new Glavnaya();

f1.Show();

this.Close();

}

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

{

String roomnum = textBox1.Text;

String FIO = textBox2.Text;

String status = textBox3.Text;

string query = "UPDATE rooms SET Room\_num = '" + roomnum + "', FIO = '" + FIO + "', Status = '" + status + "', Vyselenie\_date = 0 WHERE Treaty\_num = '"+ Izmenenie.izm\_num +"'";

string query1 = "UPDATE archive SET Room\_num = '" + roomnum + "', FIO = '" + FIO + "', Status = '" + status + "', Vyselenie\_date = 0 WHERE Treaty\_num = '" + Izmenenie.izm\_num + "'";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlCommand cmd1 = new MySqlCommand(query1, conn);

cmd1.ExecuteNonQuery();

cmd.ExecuteNonQuery();

}

}

}