

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | Системы автоматизированного проектирования | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Разработка программных систем | | |
| Курс | II | | | Группа | 423 |

Отчёт по контрольной работе №4

Вариант № 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнитель: |  |  |  |  |
| обучающийся группы 423 |  |  |  | Лянко Максим Владимирович |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Корниенко Иван Григорьевич |
|  |  | (дата, подпись) |  | Макарук Роман Валерьевич |
|  |  |  |  | Федин Алексей Константинович |

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc134647499)

[2 Исходные данные 3](#_Toc134647500)

[3 Особые ситуации 3](#_Toc134647501)

[4 Математические методы и алгоритмы решения задач 4](#_Toc134647502)

[5 Форматы представления данных 4](#_Toc134647503)

[6 Структура программы 5](#_Toc134647504)

[7 Блок-схема алгоритма решения задачи 9](#_Toc134647505)

[8 Описание хода выполнения 13](#_Toc134647506)

[9 Результаты работы программы 14](#_Toc134647507)

[10 Выводы по заданию 16](#_Toc134647508)

[11 Исходный код полученного программного решения 16](#_Toc134647509)

## 1 Постановка задачи

Необходимо написать приложение с использованием технологии WinForms реализующие вариант задания. Программа должна позволять добавлять новые сущности с использованием интерфейса и редактировать существующие. Сущности, добавленные в программу, должны сохраняться между запусками приложения. Для хранения данных необходимо использовать СУБД SqLite. Необходимо предусмотреть возможность сохранения списка существующих сущностей в файл.

Тематика базы данных: хранение результатов спортивных соревнований

## 2 Исходные данные

Исходные данные состоят из базы данных списка результатов спортивных соревнований. Пользователь может добавлять новые сущности и редактировать уже существующие.

Поля таблицы базы данных: id, название команды, количество участников, место, дата проведения соревнований.

## 3 Особые ситуации

Необходимо рассмотреть следующие особые ситуации:

* программа не дает пользователю вводить никакие символы, кроме цифр, в поля, запрашивающие числа
* программа не дает пользователю вводить никакие символы, кроме букв, в поля, запрашивающие строки
* программа не дает пользователю изменить сущности, с несуществующим в БД id

## 4 Математические методы и алгоритмы решения задач

Согласно постановке задачи, для получения необходимых результатов будут использоваться операторы “>”, “≥”, “<”, “≤”, “=”, “+”.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 Форматы представления данных Таблица 1 - Основные переменные, используемые в программе   |  |  |  | | --- | --- | --- | | Имя | Тип | Описание | | connection | SqliteConnection | Объект ддя подключения к базе данных | | command | SqliteCommand | Объект основых комманд для работы с базой данных |    6 Структура программы Таблица 2 - Модули программы   |  |  | | --- | --- | | Название модуля | Описание | | Program | Запуск основной формы приложения | | Form1 | Главная форма | | Form2 | Логика и структура формы, для добавления новой сущности в БД | | Form3 | Логика и структура формы, для просмотра и удаления существующих сущностей в БД | | Form4 | Логика и структура формы, для редактирования существующей сущности в БД | |  |

Таблица 3 – Основные функции модуля Form1

|  |  |
| --- | --- |
| Имя | Описание |
| addButtonClick | Открытие формы с функцией добавления сущностей в базу данных |
| showListButtonClick | Открытие формы с просмотром существующей сущности |

Таблица 4 – Функции модуля Form2

|  |  |
| --- | --- |
| Имя | Описание |
| checkIfDataExists | Добавление сущностей в базу данных |

Таблица 5 – Функции модуля Form3

|  |  |
| --- | --- |
| Имя | Описание |
| showList | Отображение всех сузностей из базы днных |
| saveOnFile | Сохранение существующих сущностей в файл |
| openEditForm | Открытие формы редактирования |
| button1\_Click | Обработка нажатия кнопки и удаление сущности из БД |
| dataGridView1\_CellDoubleClick | Выбор элемента по двойному клику |

Таблица 6 – Функции модуля Form4

|  |  |
| --- | --- |
| Имя | Описание |
| searchResultById | Поиск сущности по его ID |
| saveChanges | Изменение данных сущности |

## 7 Блок-схема алгоритма решения задачи

На рисунке 1 представлена блок-схема работы программы.



Рисунок 1 – Блок-схема работы программы

## 8 Описание хода выполнения

При запуске программы открывается форма со списком сущностей базы данных и боковым меню. Нажимая на кнопки бокового меню, пользователь может перемещаться между формами добавления и редактирования сущностей. В форме списка пользователь может сохранить сущности из базы данных в формате json. При двойном нажатии на одно из полей сущности открывается форма редактирования сущности. В форме редактирования сущности пользователь может изменить поля, удалить сущность или ничего не делать. При нажатии кнопки «Apply changes» программа изменит сущность в базе данных, после чего откроется форма со списком.

## 9 Результаты работы программы

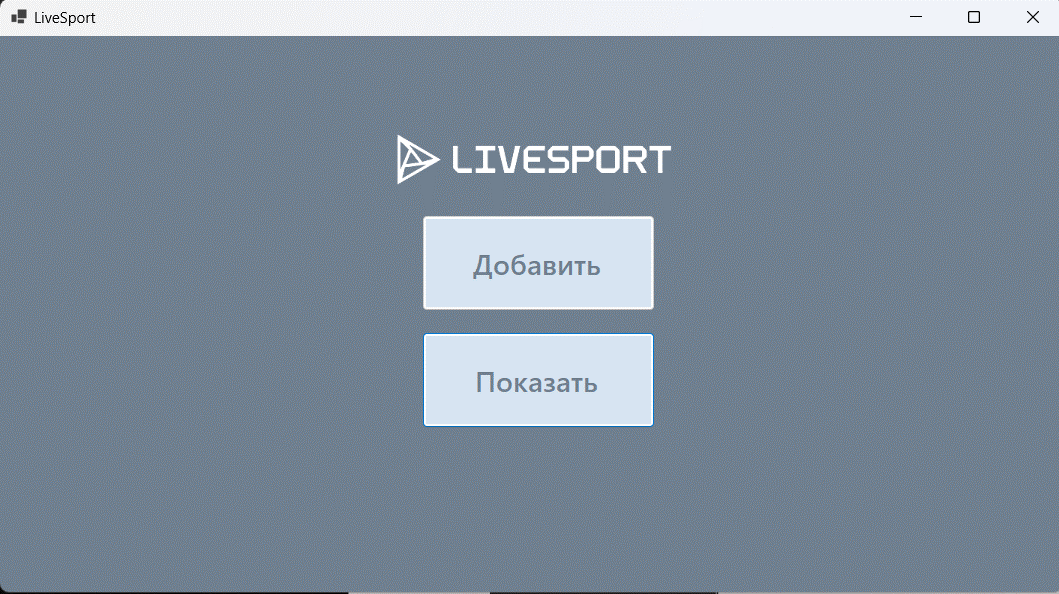


Рисунок 2 – Экранная копия окна программы сразу после запуска

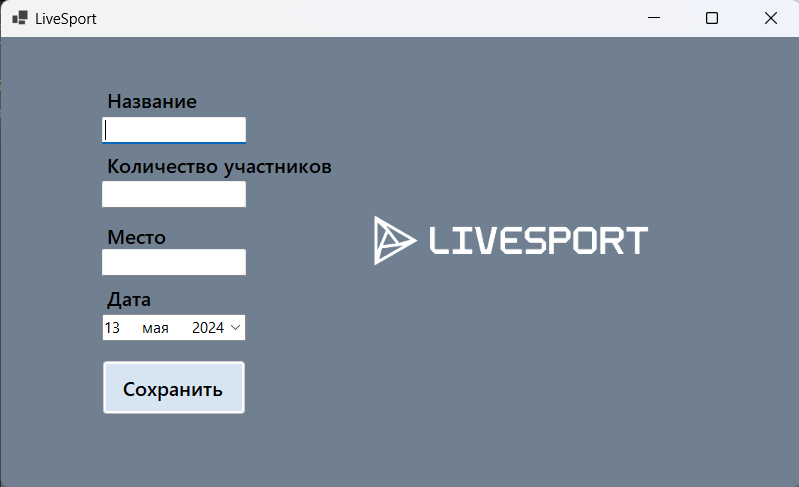


Рисунок 3 – Экранная копия окна программы после открытия формы добавления новой сущности

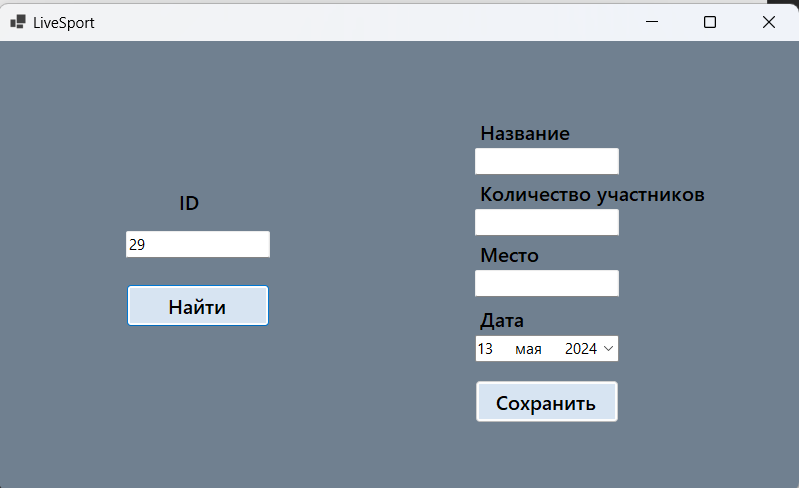


Рисунок 4 – Экранная копия окна программы после открытия формы изменения сущности

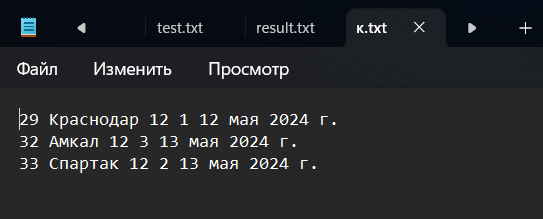


Рисунок 4 – Экранная копия файла с сохраненными сущностями

## 10 Выводы по заданию

В ходе выполнения задания были изучены основы работы с языком С# и работы с системой WinForms. Были получены навыки создания приложений с полноценным графическим интерфейсом, а также навыки взаимодействия с базой данных SQLite.

## 11 Исходный код полученного программного решения

Файл Program.cs:

using Microsoft.Data.Sqlite;

namespace FourthLab

{

internal static class Program

{

[STAThread]

static void Main()

{

ApplicationConfiguration.Initialize();

Application.Run(new Form1());

}

}

}

Файл Form1.cs:

namespace FourthLab

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

MessageBox.Show("\t М. В. Лянко, группа 423, СПбГТИ \n \tКонтрольная работа #4, вариант 10 \n");

}

private void addButtonClick(object sender, EventArgs e)

{

Form2 newForm = new Form2();

newForm.Show();

}

private void showListButtonClick(object sender, EventArgs e)

{

Form3 newForm = new Form3();

newForm.Show();

MessageBox.Show("Для редактирования нажмите два раза по элементу");

newForm.showList();

}

}

}

Файл Form2.cs:

using Microsoft.Data.Sqlite;

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;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace FourthLab

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

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

{

checkIfDataExists();

}

private void checkIfDataExists()

{

string dbPath = "Data Source=C:\\Users\\Максим\\Desktop\\4 сем\\РПС\\FourthLab\\FourthLab\\DataBase.db";

try

{

if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) || string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(dateTimePicker1.Text))

{

MessageBox.Show("[!] Заполните все поля");

return;

}

string name = textBox1.Text;

if (!int.TryParse(textBox2.Text, out int numberOfPlayers) || numberOfPlayers <= 0)

{

MessageBox.Show("Введите корректное количество участников.");

return;

}

if (!int.TryParse(textBox3.Text, out int place) || place <= 0)

{

MessageBox.Show("Введите корректное место.");

return;

}

string date = dateTimePicker1.Text;

using (SqliteConnection connection = new SqliteConnection(dbPath))

{

connection.Open();

string query = "INSERT INTO Results(name, numberOfPlayers, place, date) VALUES (@Name, @NumberOfPlayers, @Place, @Date)";

SqliteCommand command = new SqliteCommand(query, connection);

command.Parameters.AddWithValue("@Name", name);

command.Parameters.AddWithValue("@NumberOfPlayers", numberOfPlayers);

command.Parameters.AddWithValue("@Place", place);

command.Parameters.AddWithValue("@Date", date);

command.ExecuteNonQuery();

connection.Close();

}

MessageBox.Show("Данные успешно сохранены");

Close();

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

}

}

Файл Form3.cs:

using Microsoft.Data.Sqlite;

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;

using static System.ComponentModel.Design.ObjectSelectorEditor;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace FourthLab

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

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

{

saveOnFile();

}

public void showList()

{

string dbPath = "Data Source=C:\\Users\\Максим\\Desktop\\4 сем\\РПС\\FourthLab\\FourthLab\\DataBase.db";

try

{

using (SqliteConnection connection = new SqliteConnection(dbPath))

{

connection.Open();

using (SqliteCommand command = new SqliteCommand("SELECT \* FROM Results", connection))

{

using (SqliteDataReader reader = command.ExecuteReader())

{

dataGridView1.Rows.Clear();

while (reader.Read())

{

dataGridView1.Rows.Add(reader.GetInt32(0), reader.GetString(1), reader.GetInt32(2), reader.GetInt32(3), reader.GetString(4));

}

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

private void saveOnFile()

{

int rowCounter = dataGridView1.RowCount;

int columnCount = dataGridView1.ColumnCount;

string[] line = new string[columnCount];

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "txt file (\*.txt)|\*.txt";

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

{

using (StreamWriter writer = new StreamWriter(sfd.FileName))

{

for (int i = 0; i < dataGridView1.RowCount; i++)

{

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

line[j] = (dataGridView1.Rows[i].Cells[j].Value ?? "").ToString();

}

writer.WriteLine(string.Join(" ", line));

}

}

}

}

private void openEditForm(int id)

{

Form4 newForm = new Form4(id);

newForm.Show();

}

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

{

if (dataGridView1.SelectedRows.Count > 0)

{

DataGridViewRow selectedRow = dataGridView1.SelectedRows[0];

int idToDelete = Convert.ToInt32(selectedRow.Cells["newColumn1"].Value);

using (SqliteConnection connection = new SqliteConnection("Data Source=C:\\Users\\Максим\\Desktop\\4 сем\\РПС\\FourthLab\\FourthLab\\DataBase.db"))

{

connection.Open();

string sql = "DELETE FROM Results WHERE ID = @id";

using (SqliteCommand command = new SqliteCommand(sql, connection))

{

command.Parameters.AddWithValue("@id", idToDelete);

command.ExecuteNonQuery();

}

}

dataGridView1.Rows.Remove(selectedRow);

}

else

{

MessageBox.Show("Выберите строку для удаления.");

}

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

DataGridViewRow row = dataGridView1.Rows[e.RowIndex];

int id = Convert.ToInt32(row.Cells["newColumn1"].Value);

openEditForm(id);

}

}

}

}

Файл Form4.cs:

using Microsoft.Data.Sqlite;

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;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace FourthLab

{

public partial class Form4 : Form

{

private int id;

public Form4(int id)

{

InitializeComponent();

this.id = id;

textBox1.Text = id.ToString();

}

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

{

searchResultById();

}

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

{

saveChanges();

}

public void searchResultById()

{

string dbPath = "Data Source=C:\\Users\\Максим\\Desktop\\4 сем\\РПС\\FourthLab\\FourthLab\\DataBase.db";

try

{

using (SqliteConnection connection = new SqliteConnection(dbPath))

{

connection.Open();

string query = $"SELECT \* FROM Results WHERE id={id};";

using (SqliteCommand command = new SqliteCommand(query, connection))

{

using (SqliteDataReader reader = command.ExecuteReader())

{

if (reader.Read())

{

textBox2.Text = reader.GetString(1);

textBox3.Text = reader.GetInt32(2).ToString();

textBox4.Text = reader.GetInt32(3).ToString();

dateTimePicker1.Text = reader.GetString(4);

}

else

{

MessageBox.Show("Результат с таким ID не найден");

}

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

private void saveChanges()

{

string dbPath = "Data Source=C:\\Users\\Максим\\Desktop\\4 сем\\РПС\\FourthLab\\FourthLab\\DataBase.db";

if (!int.TryParse(textBox1.Text, out int resultID))

{

MessageBox.Show("Некорректное значение ID результата.");

return;

}

string name = textBox2.Text;

if (string.IsNullOrEmpty(name))

{

MessageBox.Show("Введите имя.");

return;

}

if (!int.TryParse(textBox3.Text, out int numberOfPlayers))

{

MessageBox.Show("Некорректное значение количества игроков.");

return;

}

if (!int.TryParse(textBox4.Text, out int place))

{

MessageBox.Show("Некорректное значение места.");

return;

}

string date = dateTimePicker1.Text;

try

{

using (SqliteConnection connection = new SqliteConnection(dbPath))

{

connection.Open();

string query = "UPDATE Results SET name=@Name, numberOfPlayers=@NumberOfPlayers, place=@Place, date=@Date WHERE id=@ResultID";

SqliteCommand command = new SqliteCommand(query, connection);

command.Parameters.AddWithValue("@Name", name);

command.Parameters.AddWithValue("@NumberOfPlayers", numberOfPlayers);

command.Parameters.AddWithValue("@Place", place);

command.Parameters.AddWithValue("@Date", date);

command.Parameters.AddWithValue("@ResultID", resultID);

command.ExecuteNonQuery();

MessageBox.Show("Изменения сохранены успешно.");

connection.Close();

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

}

}