Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

*ЛАБОРАТОРНАЯ*

*РАБОТА №1*

По Системы управления базами данных

Тема Работа с СУБД SQLite

Руководитель

Колпаков А.А.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Ерсилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2023

**Лабораторная работа №1**

Тема: Работа с СУБД SQLite

Цель работы: Приобрести практические навыки с работой СУБД SQLite

**Ход работы:**

**Задание 1**

Создать СУБД SQLite на платформе Visual Studio используя язык C#

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SQLite;

using Finisar.SQLite;

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

namespace CAPLab1

{

public partial class Form1 : Form

{

private sqliteclass myDataBase = null; //таблица базы данных

private string sCurDir = string.Empty;

private string sPath = string.Empty;//путь и имя базы данных

private string sSql = string.Empty; //запрос

public SelectdCell selCell = new SelectdCell(); //Поля выделенной ячейки

public struct SelectdCell

{

public int SelNumberRoute;

public int SelBusNumber;

public int SelDistance;

public int SelTime;

}

public Form1()

{

InitializeComponent();

}

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

{

sPath = Path.Combine(Application.StartupPath, "myDataBase.db");

Text = sPath;

}

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

{

myDataBase = new sqliteclass();

sSql = @"CREATE TABLE if not exists [infoBus]([number\_route] PRIMARY KEY NOT NULL,[number\_bus] TEXT NOT NULL,[km] INTEGER NOT NULL,[time] INTEGER NOT NULL);";

//Пытаемся создать таблицу

myDataBase.iExecuteNonQuery(sPath, sSql, 0);

sSql = @"insert into infoBus (number\_route,number\_bus,km,time) values('ааа','ааа',1,'ааа');";

//Проверка работы

if (myDataBase.iExecuteNonQuery(sPath, sSql, 1) == 0)

{

Text = "Ошибка проверки таблицы на запись, таблица или не создана или не прошла запись тестовой строки!";

myDataBase = null;

return;

}

sSql = "select \* from infoBus";

DataRow[] datarows = myDataBase.drExecute(sPath, sSql);

if (datarows == null)

{

Text = "Ошибка проверки таблицы на чтение!";

myDataBase = null;

return;

}

Text = "";

foreach (DataRow dr in datarows)

{

Text += dr["number\_route"].ToString().Trim() + dr["number\_bus"].ToString().Trim() + dr["km"].ToString().Trim() + dr["time"].ToString().Trim() + " ";

}

sSql = "delete from infoBus";

if (myDataBase.iExecuteNonQuery(sPath, sSql, 1) == 0)

{

Text = "Ошибка проверки таблицы на удаление записи!";

myDataBase = null;

return;

}

Text = "Таблица была создана!";

button2.Enabled = true;

myDataBase = null;

return;

}

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

{

myDataBase = new sqliteclass();

Form2 form2 = new Form2();

form2.ShowDialog();

if (form2.finish == true)

{

int number\_route = form2.numberRoute;

int number\_bus = form2.numberBus;

int km = form2.km;

int time = form2.time;

sSql = $@"insert into infoBus (numberRoute,numberBus,km,time) values('{number\_route}','{number\_bus}','{km}','{time}');";

//Проверка работы

try

{

if (myDataBase.iExecuteNonQuery(sPath, sSql, 1) == 0)

{

Text = "ОШИБКА ЗАПИСИ!";

}

myDataBase = null;

Text = "Запись 1 добавлена!";

}

catch (Exception)

{

MessageBox.Show("Ошибка! Форма была заполнена некорректно");

}

button4.Enabled = true;

button5.Enabled = true;

}

Read();

return;

}

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

{

myDataBase = new sqliteclass();

Form3 form3 = new Form3(selCell.SelNumberRoute, selCell.SelBusNumber, selCell.SelDistance, selCell.SelTime);

form3.ShowDialog();

if (form3.finish == true)

{

sSql = $@"Update infoBus set number\_route = '{form3.numberRoute}, number\_bus = '{form3.numberBus}', km = '{form3.km}', time = '{form3.time}';";

//Проверка работы

if (myDataBase.iExecuteNonQuery(sPath, sSql, 1) == 0)

{

Text = "Произошла ошибка обновления записи!";

myDataBase = null;

return;

}

myDataBase = null;

Text = "Запись была исправлена!";

}

Read();

}

private void button5\_Click(object sender, EventArgs e) //Clear

{

myDataBase = new sqliteclass();

sSql = "delete from infoBus";

if (myDataBase.iExecuteNonQuery(sPath, sSql, 1) == 0)

{

Text = "Произошла ошибка удаления записи!";

myDataBase = null;

return;

}

myDataBase = null;

Text = "Записи были удалены из БД!";

Read();

return;

}

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

{

if (e.RowIndex > -1)

{

selCell.SelNumberRoute = int.Parse(dataGridView1.Rows[e.RowIndex].Cells["numberRoute"].FormattedValue.ToString());

selCell.SelBusNumber = int.Parse(dataGridView1.Rows[e.RowIndex].Cells["numberBus"].FormattedValue.ToString());

selCell.SelDistance = int.Parse(dataGridView1.Rows[e.RowIndex].Cells["km"].FormattedValue.ToString());

selCell.SelTime = int.Parse(dataGridView1.Rows[e.RowIndex].Cells["time"].FormattedValue.ToString());

}

}

private void Read() //Чтение

{

myDataBase = new sqliteclass();

sSql = "select \* from infoBus";

DataRow[] datarows = myDataBase.drExecute(sPath, sSql);

if (datarows == null)

{

Text = "Ошибка чтения!";

myDataBase = null;

return;

}

Text = "";

dataGridView1.Rows.Clear();

foreach (DataRow dr in datarows)

{

dataGridView1.Rows.Add(dr["number\_route"], dr["number\_bus"], dr["km"], dr["time"]);

}

button3.Enabled = true;

}

}

}

Рисунок 1 – исходный код класса Form1.cs

using System;

using System.Windows.Forms;

namespace CAPLab1

{

public partial class Form2 : Form

{

public int numberRoute;

public int numberBus;

public int km;

public int time;

public bool finish = false;

public Form2()

{

InitializeComponent();

}

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

{

if (textBox1.Text == "")

{

MessageBox.Show("Enter field of 'bus number'");

Close();

}

else if (textBox2.Text == "")

{

MessageBox.Show("Enter field of 'bus rouse'");

Close();

}

else if (textBox3.Text == "")

{

MessageBox.Show("Enter field of 'distance'");

Close();

}

else if (textBox4.Text == "")

{

MessageBox.Show("Enter field of 'time'");

Close();

}

else

{

numberRoute = int.Parse(textBox1.Text);

numberBus = int.Parse(textBox2.Text);

km = int.Parse(textBox3.Text);

time = int.Parse(textBox4.Text);

finish = true;

this.Close();

}

}

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

{

this.Close();

}

}

}

Рисунок 2 – исходный код класса Form2.cs

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;

namespace CAPLab1

{

public partial class Form3 : Form

{

public int numberRoute;

public int numberBus;

public int km;

public int time;

public bool finish = false;

public Form3()

{

InitializeComponent();

}

public Form3(int \_number\_route, int \_number\_bus, int \_km, int \_time)

{

this.numberRoute = \_number\_route;

this.numberBus = \_number\_bus;

this.km = \_km;

this.time = \_time;

InitializeComponent();

}

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

{

textBox1.Text = $"{numberRoute}";

textBox2.Text = $"{numberBus}";

textBox3.Text = $"{km}";

textBox4.Text = $"{time}";

}

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

{

if (textBox1.Text == "")

{

MessageBox.Show("Enter field of 'bus number'");

Close();

}

else if (textBox2.Text == "")

{

MessageBox.Show("Enter field of 'bus rouse'");

Close();

}

else if (textBox3.Text == "")

{

MessageBox.Show("Enter field of 'distance'");

Close();

}

else if (textBox4.Text == "")

{

MessageBox.Show("Enter field of 'time'");

Close();

}

else

{

numberRoute = int.Parse(textBox1.Text);

numberBus = int.Parse(textBox2.Text);

km = int.Parse(textBox3.Text);

time = int.Parse(textBox4.Text);

finish = true;

this.Close();

}

}

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

{

this.Close();

}

}

}

Рисунок 3 – исходный код класса Form3.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System.Data;

using Finisar.SQLite;

namespace CAPLab1

{

internal class sqliteclass

{

//Конструктор

public sqliteclass()

{

}

public int iExecuteNonQuery(string FileData, string sSql, int where)

{

int n = 0;

try

{

using (SQLiteConnection con = new SQLiteConnection())

{

if (where == 0)

{

con.ConnectionString = @"Data Source=" + FileData + ";New=True;Version=3";

}

else

{

con.ConnectionString = @"Data Source=" + FileData + ";New=False;Version=3";

}

con.Open();

using (SQLiteCommand sqlCommand = con.CreateCommand())

{

sqlCommand.CommandText = sSql;

n = sqlCommand.ExecuteNonQuery();

}

con.Close();

}

}

catch (Exception)

{

n = 0;

}

return n;

}

public DataRow[] drExecute(string FileData, string sSql)

{

DataRow[] datarows = null;

SQLiteDataAdapter dataadapter = null;

DataSet dataset = new DataSet();

DataTable datatable = new DataTable();

try

{

using (SQLiteConnection con = new SQLiteConnection())

{

con.ConnectionString = @"Data Source=" + FileData +

";New=False;Version=3";

con.Open();

using (SQLiteCommand sqlCommand = con.CreateCommand())

{

dataadapter = new SQLiteDataAdapter(sSql, con);

dataset.Reset();

dataadapter.Fill(dataset);

datatable = dataset.Tables[0];

datarows = datatable.Select();

}

con.Close();

}

}

catch (Exception)

{

datarows = null;

}

return datarows;

}

}

}

Рисунок 4 – исходный код класса sqliteclass.cs

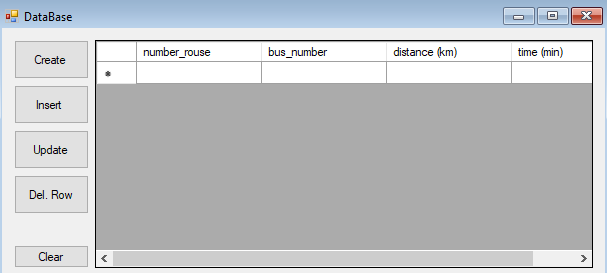


Рисунок 5 – вид конструктора формы: Form1

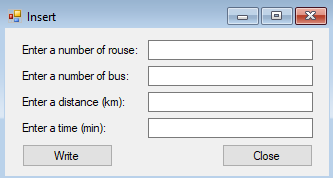


Рисунок 6 – вид конструктора формы: Form2

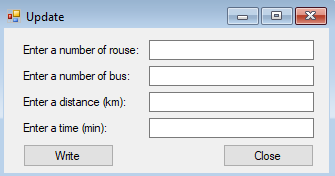


Рисунок 7 – вид конструктора формы: Form3

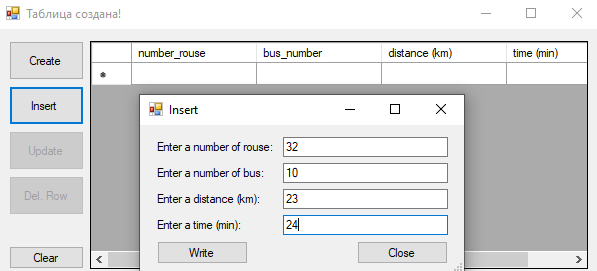


Рисунок 8 – заполнение DataBase (1/2)

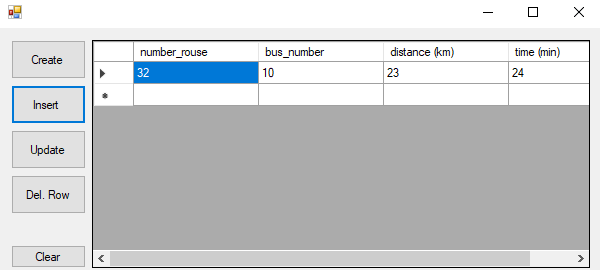


Рисунок 9 – заполнение DataBase (2/2)

Вывод: на данной лабораторной работе были приобретены навыки работы с СУБД SQLite.