

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 №2

По \_\_\_\_\_ Системам управления базами данных \_\_\_\_\_

Тема \_\_\_\_\_ Работа с изображением \_\_\_\_\_

Руководитель

\_\_\_\_\_ Колпаков А.А. \_\_\_\_\_

(фамилия, инициалы)

\_\_\_\_\_ (подпись)

\_\_\_\_\_ (дата)

Студент \_\_\_\_\_ ПИН – 121 \_\_\_\_\_

(группа)

\_\_\_\_\_ Ермилов М.В. \_\_\_\_\_

(фамилия, инициалы)

\_\_\_\_\_ (подпись)

\_\_\_\_\_ (дата)

Муром 2023

## Лабораторная работа № 2

**Тема работы:** Работа с изображением

**Цель работы:** Получение навыков в работе с изображением в SQLITE.

**Ход работы:**

Листинг 1 class Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LB_2
{
    internal static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new DataBase());
        }
    }
}
```

Листинг 2 class sqliteclass:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlTypes;
using System.Data;
using Devart.Data;
using Devart.Data.SQLite;
using System.IO;

namespace LB_2
{
    class sqliteclass
    {
        private string sLasterror = string.Empty;
        private string sErrors = string.Empty;
        public string sGetErrors()
        {
            return sErrors;
        }
        public string sGetLastError()
        {
            return sLasterror;
        }
    }
    //Конструктор
```

					МИВУ 09.03.04 - 12.000		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.					Работа с изображением	Лит.	Лист
Провер.		Колпаков А.А.					2
Реценз.							14
Н. Контр.						МИ ВлГУ ПИН-121	
Утверд.							

```

public sqliteclass() {
}
#region iExecuteNonQuery
public int iExecuteNonQuery(string FileData, string sSql, int where)
{
    int n = 0;
    try
    {
        //File.AppendAllText("logerrors.txt", DateTime.Now.ToString() + " " +
        Path.GetFileName(FileData) + " " + sSql + " " + ex.Message + " " +
        Environment.NewLine, System.Text.Encoding.UTF8);
        using (SQLiteConnection con = new SQLiteConnection())
        {
            if (where == 0)
            {
                SQLiteConnection.CreateFile(FileData);
                con.ConnectionString = @"Data Source=" + FileData;
            }
            else
            {
                con.ConnectionString = @"Data Source=" + FileData;
            }
            con.Open();
            using (SQLiteCommand sqlCommand = con.CreateCommand())
            {
                sqlCommand.CommandText = sSql;
                n = sqlCommand.ExecuteNonQuery();
            }
            con.Close();
        }
    }
    catch (Exception ex) {
        n = 0;
        sLastError = ex.Message;
        sErrors += ex.Message + "Ошибка!!!";
    }
    return n;
}
#endregion
#region drExecute
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;
            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 ex) {
        datarows = null;
        sLastError = ex.Message;
        sErrors += ex.Message + "Ошибка!!!";
    }
}

```

					МИВУ 09.03.04 – 12.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

```

    }
    return datarows;
}
#endregion
}
}

```

Листинг 3 class DataBase:

```

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.Drawing.Imaging;
using System.Windows.Forms;
using System.Data.SQLite;
using Finisar.SQLite;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.TaskbarClock;

namespace LB_2
{
    public partial class DataBase : Form
    {
        private sqliteclass mydb = 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 sc = new SelectdCell(); //Поля выделенной ячейки

        public struct SelectdCell
        {
            public int SelAddress;
            public int SelStatus;
            public int SelWorker;
            public int SelPost;
            public int SelForeman;
            public int SelDuty;
        }

        public DataBase()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                sPath = Path.Combine(Application.StartupPath, "mybd.db");
                Text = sPath;
            }

            private void button1_Click(object sender, EventArgs e)
            {
                mydb = new sqliteclass();
                sSql = @"CREATE TABLE if not exists [construction_objects]([id]
INTEGER PRIMARY KEY AUTOINCREMENT, [address] TEXT NOT NULL,[status] TEXT NOT
NULL,[worker] TEXT NOT NULL,[post] TEXT NOT NULL,[foreman] TEXT NOT NULL,[duty]
TEXT NOT NULL);";
                mydb.iExecuteNonQuery(sPath, sSql, 0);
                sSql = @"insert into construction_objects
(address,status,worker,post,foreman,duty) values('Орловская

```

					МИВУ 09.03.04 – 12.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

```

д.25', 'Работоспособное', 'Петров Пётр Петрович', 'Каменщик', 'Иванов Иван
Иванович', 'Контролёр');";
    if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
    {
        Text = "Ошибка проверки таблицы на запись, таблица или не создана
или не прошла запись тестовой строки!";
        mydb = null;
        return;
    }
    sSql = "select * from construction_objects";
    DataRow[] datarows = mydb.drExecute(sPath, sSql);
    if (datarows == null)
    {
        Text = "Ошибка проверки таблицы на чтение!";
        mydb = null;
        return;
    }
    Text = "";
    foreach (DataRow dr in datarows)
    {
        Text += dr["address"].ToString().Trim() +
dr["status"].ToString().Trim() + dr["worker"].ToString().Trim() +
dr["post"].ToString().Trim() + dr["foreman"].ToString().Trim() +
dr["duty"].ToString().Trim() + " ";
    }

    sSql = "delete from construction_objects";
    if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
    {
        Text = "Ошибка проверки таблицы на удаление записи!";
        mydb = null;
        return;
    }

    Text = "Таблица создана!";
    button2.Enabled = true;
    mydb = null;
    return;
}
private void button2_Click(object sender, EventArgs e) //Insert
{
    mydb = new sqliteclass();
    Insert f2 = new Insert();
    f2.ShowDialog();
    if (f2.finish == true)
    {
        int address = f2._address;
        int status = f2._status;
        int worker = f2._worker;
        int post = f2._post;
        int foreman = f2._foreman;
        int duty = f2._duty;
        sSql = $"insert into construction_objects
(address,status,worker,post,foreman,duty)
values('{address}','{status}','{worker}','{post}','{foreman}','{duty}')";
        //Проверка работы
        try
        {
            if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
            {
                Text = "Ошибка записи!";
            }
            mydb = null;
            Text = "Запись 1 добавлена!";
        }
    }
}

```

					МИВУ 09.03.04 – 12.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

```

        catch (Exception)
        {
            MessageBox.Show("Ошибка! Форма заполнена некорректно");
        }
        button4.Enabled = true;
        button5.Enabled = true;
    }
    Read();

    return;
}

private void button4_Click(object sender, EventArgs e) //Update
{
    mydb = new sqliteclass();
    Update f3 = new Update(sc.SelAddress, sc.SelStatus, sc.SelWorker,
sc.SelPost, sc.SelForeman, sc.SelDuty);
    f3.ShowDialog();
    if (f3.finish == true)
    {
        sSql = $"Update construction_objects set address =
'{{f3.address}}', status = '{{f3.status}}', worker = '{{f3.worker}}', post =
'{{f3.post}}', foreman = '{{f3.foreman}}', duty = '{{f3.duty}}';";
        //Проверка работы
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка обновления записи!";
            mydb = null;
            return;
        }
        mydb = null;
        Text = "Запись исправлена!";
    }
    Read();
}

private void button5_Click(object sender, EventArgs e) //Clear
{
    mydb = new sqliteclass();
    sSql = "delete from infoBus";
    if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
    {
        Text = "Ошибка удаления записи!";
        mydb = null;
        return;
    }
    mydb = null;
    Text = "Записи удалены из БД!";
    Read();
    return;
}

private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    if (e.RowIndex > -1)
    {
        sc.SelAddress =
int.Parse(dataGridView1.Rows[e.RowIndex].Cells["address"].FormattedValue.ToString());
        sc.SelStatus =
int.Parse(dataGridView1.Rows[e.RowIndex].Cells["status"].FormattedValue.ToString());
    }
}

```

					МИВУ 09.03.04 – 12.000	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

```

        sc.SelWorker =
int.Parse(dataGridView1.Rows[e.RowIndex].Cells["worker"].FormattedValue.ToString()
);
        sc.SelPost =
int.Parse(dataGridView1.Rows[e.RowIndex].Cells["post"].FormattedValue.ToString());
        sc.SelForeman =
int.Parse(dataGridView1.Rows[e.RowIndex].Cells["foreman"].FormattedValue.ToString()
);
        sc.SelDuty =
int.Parse(dataGridView1.Rows[e.RowIndex].Cells["duty"].FormattedValue.ToString());
    }
}

private void Read() //read
{
    mydb = new sqliteclass();
    sSql = "select * from infoBus";
    DataRow[] datarows = mydb.drExecute(sPath, sSql);

    if (datarows == null)
    {
        Text = "Ошибка чтения!";
        mydb = null;
        return;
    }
    Text = "";

    dataGridView1.Rows.Clear();

    foreach (DataRow dr in datarows)
    {
        dataGridView1.Rows.Add(dr["address"], dr["status"], dr["worker"],
dr["post"], dr["foreman"], dr["duty"]);
    }
    button3.Enabled = true;
}

private void button5_Click_1(object sender, EventArgs e) //Delete Row
{
    //if (this.dataGridView1.SelectedRows.Count > 0)
    //{
    //    mydb = new sqliteclass();
    //    sSql = "DELETE from infoBus";
    //    dataGridview1.Rows.RemoveAt(this.dataGridView1.SelectedRows[0].Index);
    //    DELETE FROM Products WHERE Manufacturer = 'Huawei'
    //}

private void button6_Click(object sender, EventArgs e)
{
    ImageForm image = new ImageForm();
    image.ShowDialog();
}
}
}

```

Листинг 4 class ImageForm:

```

using System;
using System.Data;
using System.Data.SQLite;
using System.Drawing;
using System.IO;
using System.Windows.Forms;

```

					МИВУ 09.03.04 – 12.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

```

using LB_2.tools;

namespace LB_2
{
    public partial class ImageForm : Form
    {
        private sqliteclass mydb = null;
        private string sPath = string.Empty;
        private string sSql = string.Empty;
        string imgFormat;
        string imgName;

        OpenFileDialog openFileDialog1 = new OpenFileDialog();

        public ImageForm()
        {
            InitializeComponent();
        }

        public byte[] ImageToByte(Image image, System.Drawing.Imaging.ImageFormat
format)
        {
            using (MemoryStream ms = new MemoryStream())
            {
                // Convert Image to byte[]
                image.Save(ms, format);
                byte[] imageBytes = ms.ToArray();
                return imageBytes;
            }
        }

        void SaveImage(byte[] imagen)
        {
            DateTime now = DateTime.Now;
            db_connect.cmd = db_connect.conn.CreateCommand();
            db_connect.cmd.CommandText = String.Format("INSERT INTO myphoto
([name], [format], [date], [photo]) VALUES ('" + imgName + "', '" + imgFormat +
"'', '" + now + "', @0);");
            SQLiteParameter param = new SQLiteParameter("@0",
System.Data.DbType.Binary);
            param.Value = imagen;
            db_connect.cmd.Parameters.Add(param);
            db_connect.conn.Open();
            try
            {
                db_connect.cmd.ExecuteNonQuery();
                MessageBox.Show("Image uploaded");
            }
            catch (Exception exc1)
            {
                MessageBox.Show(exc1.Message);
            }
            db_connect.conn.Close();
        }

        public Image ByteToImage(byte[] imageBytes)
        {
            // Convert byte[] to Image
            MemoryStream ms = new MemoryStream(imageBytes, 0, imageBytes.Length);
            ms.Write(imageBytes, 0, imageBytes.Length);
            Image image = new Bitmap(ms);
            return image;
        }

        private void button1_Click(object sender, EventArgs e) //Create
        {
            //Создаем экземпляр нашего класса
            mydb = new sqliteclass();
        }
    }
}

```

					МИВУ 09.03.04 – 12.000	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		



```

        sSql = @"CREATE TABLE if not exists [myphoto]([id] INTEGER PRIMARY
KEY AUTOINCREMENT,[name] TEXT,[format] TEXT,[date] REAL,[photo] BLOB);";
        mydb.iExecuteNonQuery(sPath, sSql, 0);

        //Проверка работы
        sSql = @"insert into myphoto (name,format,date) values('Фамилия Имя
Отчество','Да','now')";
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка проверки таблицы на запись,";
            Text += " таблица или не создана или не прошла запись тестовой
строки!";

            mydb = null;
            return;
        }
        sSql = "select * from myphoto";
        DataRow[] datarows = mydb.drExecute(sPath, sSql);
        if (datarows == null)
        {
            Text = "Ошибка проверки таблицы на чтение!";
            mydb = null;
            return;
        }
        Text = "";
        foreach (DataRow dr in datarows)
        {
            Text += dr["id"].ToString().Trim() + dr["name"].ToString().Trim()
+ dr["format"].ToString().Trim() + " ";
        }
        sSql = "delete from myphoto";
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка проверки таблицы на удаление записи!";
            mydb = null;
            return;
        }
        Text = "Таблица создана!";
        mydb = null;
        return;
    }

    private void button5_Click(object sender, EventArgs e) //Delete
    {
        mydb = new sqliteclass();
        sSql = "delete from myphoto";
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка удаления записи!";
            mydb = null;
            return;
        }
        mydb = null;
        Text = "Записи удалены из БД!";
    }

    private void button6_Click(object sender, EventArgs e) //Update
    {
        mydb = new sqliteclass();
        sSql = "select * from myphoto";
        DataRow[] datarows = mydb.drExecute(sPath, sSql);

        if (datarows == null)
        {
            Text = "Ошибка чтения!";
            mydb = null;
            return;
        }
    }

```

```

    }
    Text = "";

    dataGridView1.Rows.Clear();

    foreach (DataRow dr in datarows)
    {
        dataGridView1.Rows.Add(dr["id"], dr["name"], dr["format"],
dr["date"]);
    }

    foreach (DataRow dr in datarows)
    {
        Text += dr["id"].ToString().Trim() + " " +
dr["name"].ToString().Trim() + " " + dr["format"].ToString().Trim() + " " +
dr["date"].ToString().Trim() + " ";
    }
    mydb = null;
}

private void button7_Click(object sender, EventArgs e) //Insert Image
{
    openFileDialog1.InitialDirectory =
@"C:\Users\eduar\OneDrive\Изображения\Saved Pictures";
    openFileDialog1.Filter = "Image
Files(*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PNG|All files (*.*)|*.*";
    //формат загружаемого файла

    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        Image photo = new Bitmap(openFileDialog1.FileName);

        string imgPath = openFileDialog1.FileName; // изображение
        imgFormat = Path.GetExtension(imgPath).Replace(".", "").ToLower();
        // запишем в переменную расширение изображения в нижнем регистре, не забыв
        // удалить точку перед расширением, получим «png»
        imgName =
Path.GetFileName(imgPath).Replace(Path.GetExtension(imgPath), ""); // запишем в
переменную имя файла, не забыв удалить расширение с точкой, получим «image-01»

        byte[] pic = ImageToByte(photo,
System.Drawing.Imaging.ImageFormat.Png);
        SaveImage(pic);
    }
}

private void button8_Click(object sender, EventArgs e) //Read Image
{
    string query = "SELECT photo FROM myphoto WHERE ID=@id;";
    db_connect.cmd = new SQLiteCommand(query, db_connect.conn);
    db_connect.cmd.Parameters.AddWithValue("@id", numericUpDown1.Value);
    db_connect.conn.Open();
    try
    {
        IDataReader rdr = db_connect.cmd.ExecuteReader();
        try
        {
            while (rdr.Read())
            {
                byte[] a = (System.Byte[])rdr[0];
                pictureBox1.Image = ByteToImage(a);
            }
        }
        catch (Exception exc) { MessageBox.Show(exc.Message); }
    }
}

```

					МИВУ 09.03.04 – 12.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

```

        catch (Exception ex) { MessageBox.Show(ex.Message); }
        db_connect.conn.Close();
    }

    private void ImageForm_Load(object sender, EventArgs e)
    {
        sPath = Path.Combine(Application.StartupPath, "mybdImage.db");
        //Чтобы не плодить MessageBox все сообщения выводим в заголовок формы
        Text = sPath;
    }
}

```

Листинг 5 class Insert:

```

using System;
using System.Windows.Forms;

namespace LB_2
{
    public partial class Insert : Form
    {
        public int _address;
        public int _status;
        public int _worker;
        public int _post;
        public int _foreman;
        public int _duty;
        public bool finish = false;

        public Insert()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "")
            {
                MessageBox.Show("Адрес:");
                Close();
            }
            else if (comboBox1.Text == "")
            {
                MessageBox.Show("Статус:");
                Close();
            }
            else if (textBox3.Text == "")
            {
                MessageBox.Show("Рабочий(ФИО):");
                Close();
            }
            else if (comboBox2.Text == "")
            {
                MessageBox.Show("Должность:");
                Close();
            }
            else if (textBox5.Text == "")
            {
                MessageBox.Show("Прораб(ФИО):");
                Close();
            }
            else if (textBox6.Text == "")
            {

```

```

        MessageBox.Show("Обязанность:");
        Close();
    }
    else
    {
        _address = int.Parse(textBox1.Text);
        _status = int.Parse(comboBox1.Text);
        _worker = int.Parse(textBox3.Text);
        _post = int.Parse(comboBox2.Text);
        _foreman = int.Parse(textBox5.Text);
        _duty = int.Parse(textBox6.Text);
        finish = true;
        this.Close();
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

Листинг 6 class Update:

```

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 LB_2
{
    public partial class Update : Form
    {
        public int _address;
        public int _status;
        public int _worker;
        public int _post;
        public int _foreman;
        public int _duty;
        public bool finish = false;
        internal object address;
        internal object status;
        internal object worker;
        internal object post;
        internal object foreman;
        internal object duty;

        public Update()
        {
            InitializeComponent();
        }
        public Update(int _address, int _status, int _worker, int _post, int
_foreman, int _duty)
        {
            this._address = _address;
            this._status = _status;
            this._worker = _worker;
            this._post = _post;

```

					МИВУ 09.03.04 – 12.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

```

        this._foreman = _foreman;
        this._duty = _duty;
        InitializeComponent();
    }

    private void Form3_Load(object sender, EventArgs e)
    {
        textBox1.Text = $"{_address}";
        comboBox1.Text = $"{_status}";
        textBox3.Text = $"{_worker}";
        comboBox2.Text = $"{_post}";
        textBox5.Text = $"{_foreman}";
        textBox6.Text = $"{_duty}";
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (textBox1.Text == "")
        {
            MessageBox.Show("Адрес:");
            Close();
        }
        else if (comboBox1.Text == "")
        {
            MessageBox.Show("Статус:");
            Close();
        }
        else if (textBox3.Text == "")
        {
            MessageBox.Show("Рабочий(ФИО):");
            Close();
        }
        else if (comboBox2.Text == "")
        {
            MessageBox.Show("Должность:");
            Close();
        }
        else if (textBox5.Text == "")
        {
            MessageBox.Show("Прораб(ФИО):");
            Close();
        }
        else if (textBox6.Text == "")
        {
            MessageBox.Show("Обязанность:");
            Close();
        }
        else
        {
            _address = int.Parse(textBox1.Text);
            _status = int.Parse(comboBox1.Text);
            _worker = int.Parse(textBox3.Text);
            _post = int.Parse(comboBox2.Text);
            _foreman = int.Parse(textBox5.Text);
            _duty = int.Parse(textBox6.Text);
            finish = true;
            this.Close();
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.Close(); }
    }
}

```

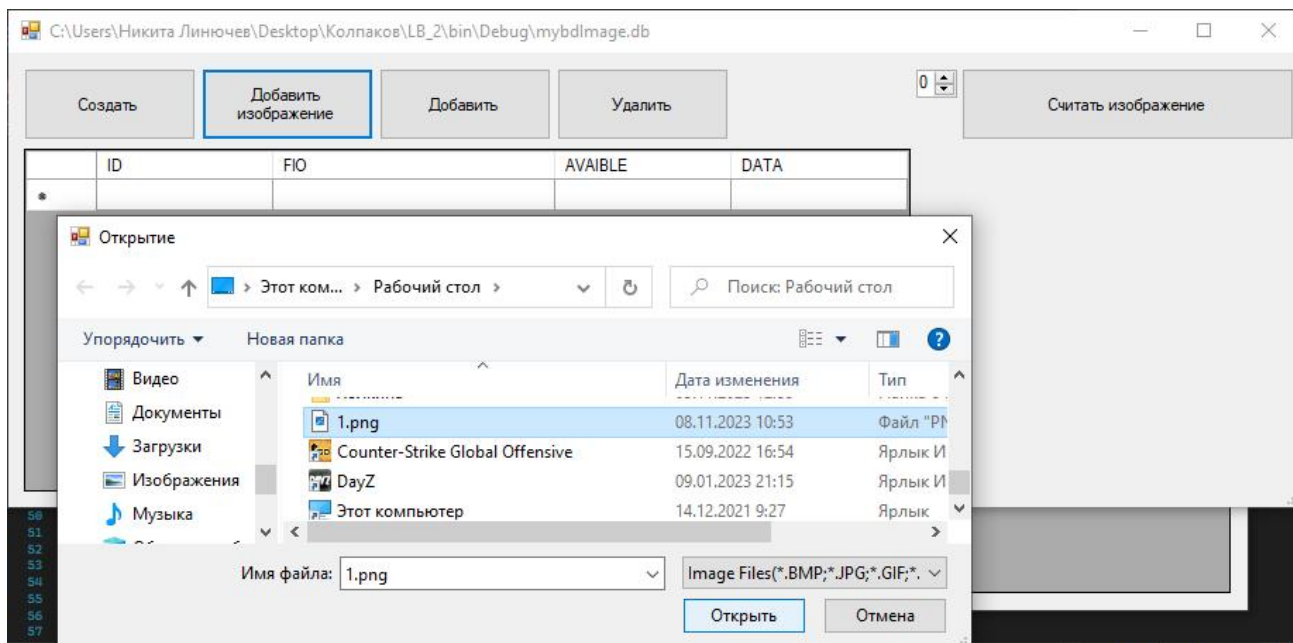


Рисунок 1 – Добавление изображение

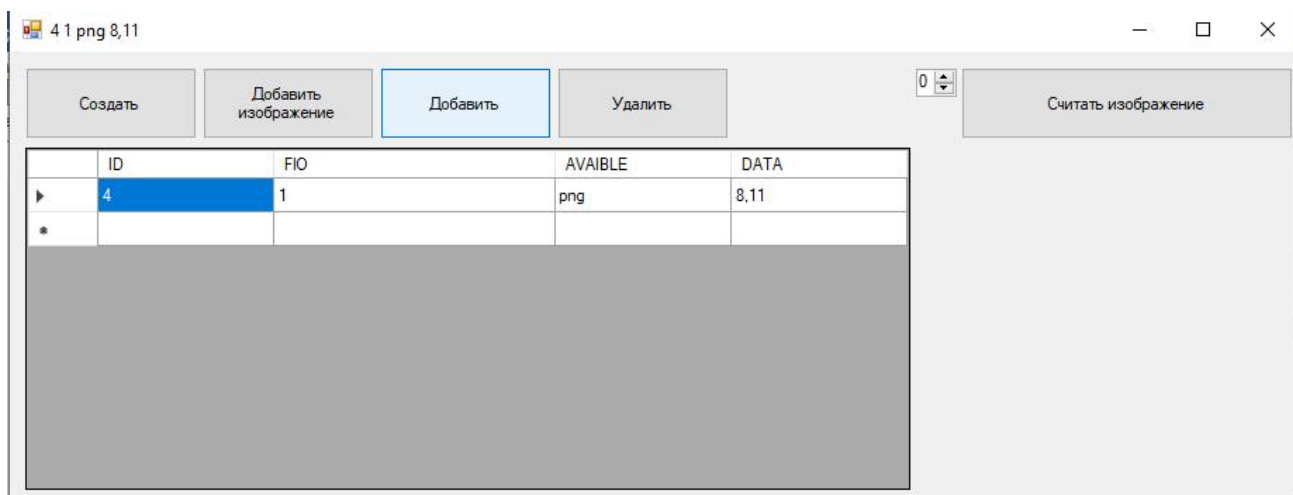


Рисунок 2 – Добавление в список

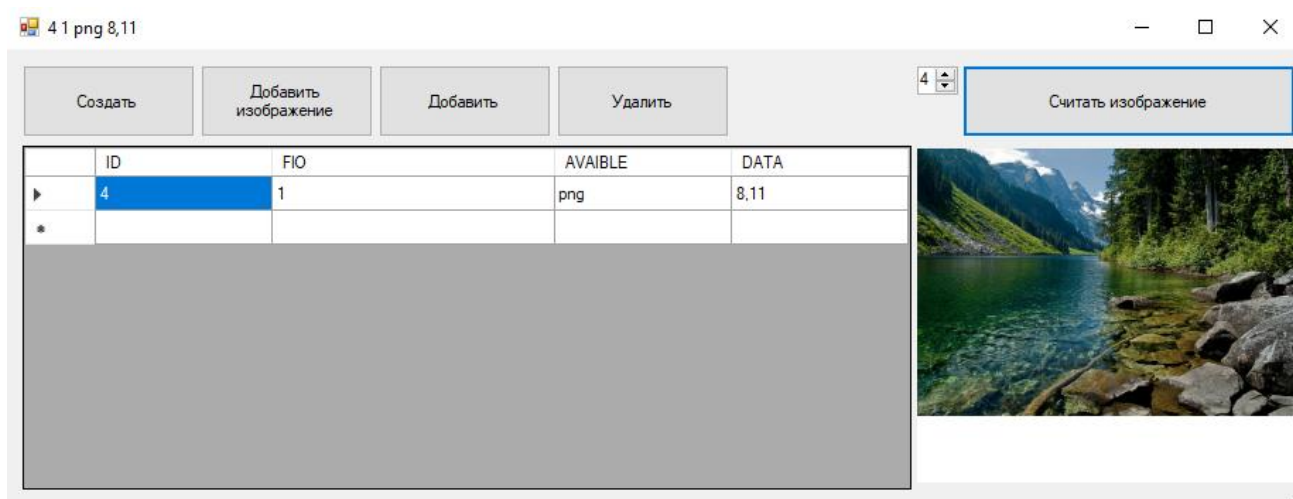


Рисунок 3 – Выбор по ID и вывод изображение

**Вывод:** В ходе лабораторной работы получены практические навыки в работе с изображением в SQLITE.