МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчёт по лабораторной работе №11

**«Работа с базами данных — XMLи OLE»**

Работу выполнил  
студент 42 группы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Б.М. Ибрагимов

Направление подготовки 01.03.02 Прикладная математика и информатика

Проверил  
канд. техн. наук, доц.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Полупанов

Краснодар 2020

**Цель работы:** является изучение работы с базами данных, ODBC, OLE, файлами XML.

**Задача 1:** ввести и выполнить приложение по указанным действиям из лабораторной. Проанализировать и объяснить работу приложений.

**ЛИСТИНГ ПРОГРАММЫ**

**Main Class**

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 System.Xml;

using System.IO;

using System.Drawing.Imaging;

namespace lab11 {

public partial class LWP10Wallpapper : Form {

private string String\_Path = String.Empty;

XmlDocument XML;

// Классы для работв с XML-документом как с объектом базы данных

DataTable WallpapperDataTable = null;

DataSet WallpapperDataSet = null;

public LWP10Wallpapper() {

InitializeComponent();

}

private void GetPicture(string X) {

DataRow[] datarows = null;

try {

datarows = WallpapperDataTable.Select("id=" + X); // Получаемвседанные DataTable поключу<id>X</id>

}

catch (Exception ex) {

MessageBox.Show(ex.Message, "Работасбазамиданных (C#) :: Базаданныхобоев");

return;

}

if (datarows.Length > 0) // Непустоли?

{

foreach (DataRow datarow in datarows) // Перебираемвсестолбцызаписи

{

L\_NAME.Text = "Имяфайла: " + datarow["name"].ToString();

string s3 = datarow["pichash"].ToString();

L\_SIZE.Text = "Разрешениерисунка: " + datarow["size"].ToString();

using (MemoryStream MS = new MemoryStream()) {

for (int i = 0; i < s3.Length / 2; i++) // Обходимполовинузнаковстрокииз<pichash>s3</pichash>вверхнемцикле, макс. i = 249 если s3.Length = 500

{

if (i \* 2 + 1 < s3.Length) // Последнееусловие: 249 \* 2 + 1 < 500

{

String s02 = Convert.ToString(s3[i \* 2]); // Чётныесимволы (посл.: 249 \* 2 = 498 символ, предпоследнийвстроке)

string s03 = Convert.ToString(s3[i \* 2 + 1]); // Нечётные символы (посл.: 249 \* 2 + 1 = 499 символ, последний в строке)

string s04 = s02 + s03; // Объединяем в одну строку (A + B)

int a = int.Parse(s04, System.Globalization.NumberStyles.HexNumber); // 32 разряда получаем из двух 16 разрядных чисел юникода (A + B)

MS.WriteByte(Convert.ToByte(a)); // Восстанавливает один байт за проход (из 32 разрядного представления знакового целого числа), всего 250 проходов по два символа за раз

}

}

PB\_MAIN.Image = Image.FromStream(MS); // Восстанавливаем рисунок (создаём рисунок из потока байтов)

}

}

}

}

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

if (TB\_NUMBER.Text.Trim() != "") {

TB\_NAME.Text = "";

TB\_FORMAT.Text = "";

TB\_SIZE.Text = "";

PB\_MAIN.Image = null;

GetPicture(TB\_NUMBER.Text.Trim()); // Отправляем номер функции, которая вытащит из XML-документа все данные

}

}

private void LWP10Wallpapper\_Load(object sender, EventArgs e) {

//Path = Directory.GetCurrentDirectory(); // Текущая директория, из которой запущено приложения

String\_Path = @"C:\Users\Ibrag\Desktop\C#\Lab Works\Programms\lab11\";

// Инициализируем объект StreamReader, считывающий символы из потока байтов в определённой кодировке

using (StreamReader SR = new StreamReader(String\_Path + @"\Wallpapper-DB.xml", System.Text.Encoding.UTF8)) {

WallpapperDataSet = new DataSet();

WallpapperDataSet.ReadXml(SR, XmlReadMode.Auto); // Считываем XML-схемуиданныев DataSet

WallpapperDataTable = WallpapperDataSet.Tables[0];

}

}

private void B\_SEARCH\_Click(object sender, EventArgs e) {

OFD\_FIND.InitialDirectory = String\_Path;

if (OFD\_FIND.ShowDialog() == DialogResult.OK) {

PB\_MAIN.Image = Image.FromFile(OFD\_FIND.FileName);

TB\_FORMAT.Text = Path.GetExtension(OFD\_FIND.FileName);

TB\_NAME.Text = Path.GetFileNameWithoutExtension(OFD\_FIND.FileName);

TB\_SIZE.Text = PB\_MAIN.Image.Width.ToString() + "x" + PB\_MAIN.Image.Height.ToString();

}

}

private void B\_SAVE\_Click(object sender, EventArgs e) {

// Сохранять не будем - если что-то не ввели

if (PB\_MAIN.Image == null) return;

if (TB\_NAME.Text == "") return;

if (TB\_FORMAT.Text == "") return;

if (TB\_SIZE.Text == "") return;

// Строки для метода SELECT в XML-документе

DataRow[] datarows = null;

// Ищем максимальное ID в DataSet (в DataTable)

string s = string.Empty;

try {

datarows = WallpapperDataTable.Select("id=max(id)");

s = datarows[0]["id"].ToString();

}

catch (Exception ex) {

MessageBox.Show(ex.Message, "Работа с базами данных (C#) :: База данных обоев");

}

if (s == "" || s == string.Empty) // Если база данных пустая, то...

{

s = "0";

}

// Для формирования строки рисунка создаём StringBuilder

StringBuilder SB = new StringBuilder();

int i = int.Parse(s) + 1; // Если база пустая, то начинаем с 1, иначе, с максимального номера + 1

// Создаём новую строку для WallpapperDataSet

DataRow datarow = WallpapperDataSet.Tables[0].NewRow(); // Формируем DataRow на основе DataSet

// Присваиваем значения столбцам строки

datarow[0] = Convert.ToString(i);

datarow[1] = TB\_NAME.Text.Trim();

// Формируемстроковоепредставлениерисунка

using (MemoryStream MS = new MemoryStream()) {

PB\_MAIN.Image.Save(MS, ImageFormat.Gif); // Сохраняемизображениевпотом MemoryStream, расширение \*.gif

byte[] b = new byte[MS.Length]; // 8-битное число (массив) длины потока в байтах

//memorystream.Read(b, 0, (int)memorystream.Length);

b = MS.GetBuffer(); // Присваиваем byte b массивбайтовпотока

s = string.Empty;

foreach (Byte zb in b) {

int a = (int)zb;

SB.Append(a.ToString("X2")); // Формируем окончательную строку (путём добавления) из данных массива байтов в шестнадцатеричном виде (X2) (шестнадцатеричное представление каждого байта рисунка)

//value = 123456789;

//Console.WriteLine(value.ToString("X"));

//Выведет: 75BCD15

//Console.WriteLine(value.ToString("X2"));

//Выведет: 75BCD15

}

datarow[2] = Convert.ToString(SB); // Отправляемвсюстрокувстолбец pichash нашейбазыданных

}

datarow[3] = TB\_FORMAT.Text.Trim();

datarow[4] = TB\_SIZE.Text.Trim();

WallpapperDataSet.Tables[0].Rows.Add(datarow); // Формируемвсюзаписьбазыданныхв DataSet

// Удаляем строку с пустыми значениями, которые при первоначальной

// загрузке были использованы для формирования схемы

if (i == 1) {

WallpapperDataSet.Tables[0].DefaultView.AllowDelete = true;

WallpapperDataSet.Tables[0].DefaultView.Delete(0);

}

PB\_MAIN.Image = null;

TB\_SIZE.Text = "";

TB\_FORMAT.Text = "";

TB\_NAME.Text = "";

// Сохраняемданные

WallpapperDataSet.WriteXml(String\_Path + @"\Wallpapper-DB.xml", XmlWriteMode.WriteSchema);

WallpapperDataSet = new DataSet();

// Вновь загружаем сохраненные данные

WallpapperDataSet.ReadXml(String\_Path + @"\Wallpapper-DB.xml", XmlReadMode.Auto);

WallpapperDataTable = WallpapperDataSet.Tables[0];

}

private void TB\_NUMBER\_KeyPress(object sender, KeyPressEventArgs e) {

// Введённые символы должны быть только цифрами, иначе ввода не будет (символ не введеётся)

if (!Char.IsDigit(e.KeyChar)) {

e.Handled = true;

}

}

}

}

**Class LWP10Wallpapper**

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 System.Xml;

using System.IO;

using System.Drawing.Imaging;

namespace lab11 {

public partial class LWP10Wallpapper : Form {

private string String\_Path = String.Empty;

XmlDocument XML;

// Классы для работв с XML-документом как с объектом базы данных

DataTable WallpapperDataTable = null;

DataSet WallpapperDataSet = null;

public LWP10Wallpapper() {

InitializeComponent();

}

private void GetPicture(string X) {

DataRow[] datarows = null;

try {

datarows = WallpapperDataTable.Select("id=" + X); // Получаемвседанные DataTable поключу<id>X</id>

}

catch (Exception ex) {

MessageBox.Show(ex.Message, "Работасбазамиданных (C#) :: Базаданныхобоев");

return;

}

if (datarows.Length > 0) // Непустоли?

{

foreach (DataRow datarow in datarows) // Перебираемвсестолбцызаписи

{

L\_NAME.Text = "Имяфайла: " + datarow["name"].ToString();

string s3 = datarow["pichash"].ToString();

L\_SIZE.Text = "Разрешениерисунка: " + datarow["size"].ToString();

using (MemoryStream MS = new MemoryStream()) {

for (int i = 0; i < s3.Length / 2; i++) // Обходимполовинузнаковстрокииз<pichash>s3</pichash>вверхнемцикле, макс. i = 249 если s3.Length = 500

{

if (i \* 2 + 1 < s3.Length) // Последнееусловие: 249 \* 2 + 1 < 500

{

String s02 = Convert.ToString(s3[i \* 2]); // Чётныесимволы (посл.: 249 \* 2 = 498 символ, предпоследнийвстроке)

string s03 = Convert.ToString(s3[i \* 2 + 1]); // Нечётные символы (посл.: 249 \* 2 + 1 = 499 символ, последний в строке)

string s04 = s02 + s03; // Объединяем в одну строку (A + B)

int a = int.Parse(s04, System.Globalization.NumberStyles.HexNumber); // 32 разряда получаем из двух 16 разрядных чисел юникода (A + B)

MS.WriteByte(Convert.ToByte(a)); // Восстанавливает один байт за проход (из 32 разрядного представления знакового целого числа), всего 250 проходов по два символа за раз

}

}

PB\_MAIN.Image = Image.FromStream(MS); // Восстанавливаем рисунок (создаём рисунок из потока байтов)

}

}

}

}

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

if (TB\_NUMBER.Text.Trim() != "") {

TB\_NAME.Text = "";

TB\_FORMAT.Text = "";

TB\_SIZE.Text = "";

PB\_MAIN.Image = null;

GetPicture(TB\_NUMBER.Text.Trim()); // Отправляем номер функции, которая вытащит из XML-документа все данные

}

}

private void LWP10Wallpapper\_Load(object sender, EventArgs e) {

//Path = Directory.GetCurrentDirectory(); // Текущая директория, из которой запущено приложения

String\_Path = @"C:\Users\Ibrag\Desktop\C#\Lab Works\Programms\lab11\";

// Инициализируем объект StreamReader, считывающий символы из потока байтов в определённой кодировке

using (StreamReader SR = new StreamReader(String\_Path + @"\Wallpapper-DB.xml", System.Text.Encoding.UTF8)) {

WallpapperDataSet = new DataSet();

WallpapperDataSet.ReadXml(SR, XmlReadMode.Auto); // Считываем XML-схемуиданныев DataSet

WallpapperDataTable = WallpapperDataSet.Tables[0];

}

}

private void B\_SEARCH\_Click(object sender, EventArgs e) {

OFD\_FIND.InitialDirectory = String\_Path;

if (OFD\_FIND.ShowDialog() == DialogResult.OK) {

PB\_MAIN.Image = Image.FromFile(OFD\_FIND.FileName);

TB\_FORMAT.Text = Path.GetExtension(OFD\_FIND.FileName);

TB\_NAME.Text = Path.GetFileNameWithoutExtension(OFD\_FIND.FileName);

TB\_SIZE.Text = PB\_MAIN.Image.Width.ToString() + "x" + PB\_MAIN.Image.Height.ToString();

}

}

private void B\_SAVE\_Click(object sender, EventArgs e) {

// Сохранять не будем - если что-то не ввели

if (PB\_MAIN.Image == null) return;

if (TB\_NAME.Text == "") return;

if (TB\_FORMAT.Text == "") return;

if (TB\_SIZE.Text == "") return;

// Строки для метода SELECT в XML-документе

DataRow[] datarows = null;

// Ищем максимальное ID в DataSet (в DataTable)

string s = string.Empty;

try {

datarows = WallpapperDataTable.Select("id=max(id)");

s = datarows[0]["id"].ToString();

}

catch (Exception ex) {

MessageBox.Show(ex.Message, "Работа с базами данных (C#) :: База данных обоев");

}

if (s == "" || s == string.Empty) // Если база данных пустая, то...

{

s = "0";

}

// Для формирования строки рисунка создаём StringBuilder

StringBuilder SB = new StringBuilder();

int i = int.Parse(s) + 1; // Если база пустая, то начинаем с 1, иначе, с максимального номера + 1

// Создаём новую строку для WallpapperDataSet

DataRow datarow = WallpapperDataSet.Tables[0].NewRow(); // Формируем DataRow на основе DataSet

// Присваиваем значения столбцам строки

datarow[0] = Convert.ToString(i);

datarow[1] = TB\_NAME.Text.Trim();

// Формируемстроковоепредставлениерисунка

using (MemoryStream MS = new MemoryStream()) {

PB\_MAIN.Image.Save(MS, ImageFormat.Gif); // Сохраняемизображениевпотом MemoryStream, расширение \*.gif

byte[] b = new byte[MS.Length]; // 8-битное число (массив) длины потока в байтах

//memorystream.Read(b, 0, (int)memorystream.Length);

b = MS.GetBuffer(); // Присваиваем byte b массивбайтовпотока

s = string.Empty;

foreach (Byte zb in b) {

int a = (int)zb;

SB.Append(a.ToString("X2")); // Формируем окончательную строку (путём добавления) из данных массива байтов в шестнадцатеричном виде (X2) (шестнадцатеричное представление каждого байта рисунка)

//value = 123456789;

//Console.WriteLine(value.ToString("X"));

//Выведет: 75BCD15

//Console.WriteLine(value.ToString("X2"));

//Выведет: 75BCD15

}

datarow[2] = Convert.ToString(SB); // Отправляемвсюстрокувстолбец pichash нашейбазыданных

}

datarow[3] = TB\_FORMAT.Text.Trim();

datarow[4] = TB\_SIZE.Text.Trim();

WallpapperDataSet.Tables[0].Rows.Add(datarow); // Формируемвсюзаписьбазыданныхв DataSet

// Удаляем строку с пустыми значениями, которые при первоначальной

// загрузке были использованы для формирования схемы

if (i == 1) {

WallpapperDataSet.Tables[0].DefaultView.AllowDelete = true;

WallpapperDataSet.Tables[0].DefaultView.Delete(0);

}

PB\_MAIN.Image = null;

TB\_SIZE.Text = "";

TB\_FORMAT.Text = "";

TB\_NAME.Text = "";

// Сохраняемданные

WallpapperDataSet.WriteXml(String\_Path + @"\Wallpapper-DB.xml", XmlWriteMode.WriteSchema);

WallpapperDataSet = new DataSet();

// Вновь загружаем сохраненные данные

WallpapperDataSet.ReadXml(String\_Path + @"\Wallpapper-DB.xml", XmlReadMode.Auto);

WallpapperDataTable = WallpapperDataSet.Tables[0];

}

private void TB\_NUMBER\_KeyPress(object sender, KeyPressEventArgs e) {

// Введённые символы должны быть только цифрами, иначе ввода не будет (символ не введеётся)

if (!Char.IsDigit(e.KeyChar)) {

e.Handled = true;

}

}

}

}

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

Рисунок 1 – чтение и изменение таблицы из выбранной базы данных

**Вывод:** язык C# предоставляет средства для работы с базами данных, например, ODBC, OLE, а также для работы с XML. Также имеется несколько методов для представления таблиц из базы данных, например, ListBox, DataGridView, XML.