Лабораторная работа №4

“ Исследование создания mdi-приложений. Сериализация объектов.

Стандартные диалоги.”

1 ЦЕЛЬ РАБОТЫ

* Изучить особенности разработки MDI-приложений в Visual Studio .Net;
* Изучить способы сохранения данных в файл и загрузки из файла;
* Освоить механизм сериализации и десериализации объектов.

2 ПОСТАНОВКА ЗАДАЧИ

1. Создать текстовый редактор Notepad C#, добавив недостающие пункты меню и функции.
2. На основании лабораторной работы 3 создать MDI-приложение. Информация в окне должна отображаться в виде таблицы. Иметь воз­можность делать выборку данных по различным критериям. Перено­сить данные из одной формы в другую.
3. Добавить формы для ввода дополнительной информации об объ­екте и фото объекта.
4. Добавить пункты меню для сохранения объектов в файл и за­грузки. При сохранении использовать стандартные диалоговые окна и механизм сериализации. В класс добавить поле «Дата создания объек­та». Поле не сериализовать, а при десериализации заново устанавли­вать по системной дате.

3 ТЕКСТ ПРОГРАММЫ

Код текстового редактора Notepad:

using System;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Windows.Forms;

namespace Notepad

{

public partial class Form1 : Form

{

string fileName = "";

bool filesave = false;

public Form1()

{

InitializeComponent();

}

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

{

try

{

Saving(fileName);

}

catch { }

}

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

{

try

{

Saving(fileName);

}

catch { }

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

textBox1.Clear();

openFileDialog1.Filter = "Text Files (\*.txt)|\*.txt";

fileName = openFileDialog1.FileName;

textBox1.Text = File.ReadAllText(fileName, Encoding.GetEncoding(1251));

Text = fileName;

}

}

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

{

saveFileDialog1.Filter = "Text Files|\*.txt";

saveFileDialog1.DefaultExt = ".txt";

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

var nameNew = saveFileDialog1.FileName;

Saving(nameNew);

if(fileName != "")

Saving(fileName);

}

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (filesave)

{

e.Cancel = false;

return;

}

if (fileName == "" && textBox1.Text != "")

e.Cancel = MessageBox.Show("Вы хотите выйти не сохранив?", "Выход", MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.No;

if (fileName != "" && textBox1.Text != File.ReadAllText(fileName))

e.Cancel = MessageBox.Show("Вы хотите выйти не сохранив?", "Выход", MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.No;

}

void Saving(string fileSaving)

{

File.WriteAllText(fileSaving, textBox1.Text, Encoding.GetEncoding(1251));

filesave = true;

}

private void toolStripButton\_Opt\_Click(object sender, EventArgs e)

{

Regex regex = new Regex(@"\s+");

textBox1.Text = regex.Replace(textBox1.Text, " ");

}

private void toolStripButton\_About\_Click(object sender, EventArgs e)

{

MessageBox.Show("Блокнот v1.0 by IlyaD", "О программе", MessageBoxButtons.OK,

MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1);

}

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

{

textBox1.SelectAll();

}

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

{

if (textBox1.CanUndo == true)

textBox1.Undo();

}

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

{

if (textBox1.SelectedText != "")

textBox1.Cut();

}

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

{

if (textBox1.SelectionLength > 0)

textBox1.Copy();

}

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

{

if (Clipboard.ContainsText())

textBox1.Paste();

}

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

{

if (textBox1.SelectionLength > 0)

textBox1.SelectedText = "";

}

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

{

textBox1.AppendText(DateTime.Now.ToString());

}

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

{

if (fontDialog1.ShowDialog() == DialogResult.OK)

textBox1.Font = fontDialog1.Font;

}

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

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

textBox1.BackColor = colorDialog1.Color;

}

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

{

if ((fileName == "" && textBox1.Text != "") || (fileName != "" && textBox1.Text != File.ReadAllText(fileName)))

{

DialogResult result = MessageBox.Show("Вы хотите сохранить файл?", "Сохранение", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

Saving(fileName);

if (result == DialogResult.Cancel)

return;

}

textBox1.Text = "";

fileName = "";

filesave = false;

}

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

{

this.Close();

}

}

}

Код основного задания:

Файл MainForm.cs

using System;

using System.IO;

using System.Linq;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

using System.Windows.Forms;

namespace WindowsFormsApp2

{

public partial class MainForm : Form

{

public int windowNumber { get; set; } = 0;

public MainForm()

{

InitializeComponent();

}

public static bool IsInt(string str)

{

foreach (char c in str)

{

if (c < '0' || c > '9')

return false;

}

return true;

}

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

{

if (ActiveMdiChild == null)

{

MessageBox.Show("Не открыто ни одной формы!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

return;

}

Table table = (Table)ActiveMdiChild;

Exam t = (Exam)table.DataGridView.CurrentRow?.DataBoundItem;

if (t != null)

{

ViewForm viewForm = new ViewForm(t);

viewForm.ShowDialog();

}

else

{

MessageBox.Show("Не выбран студент!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

}

}

private void добавитьToolStripMenuItem1\_Click(object sender, EventArgs e)

{

AddForm addForm = new AddForm();

DialogResult result = addForm.ShowDialog();

if (result == DialogResult.OK && addForm.Result != null)

{

if (ActiveMdiChild == null)

{

новаяТаблицаToolStripMenuItem\_Click(sender, e);

}

var table = (Table)ActiveMdiChild;

table?.exams.Add(addForm.Result);

}

}

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

{

if (ActiveMdiChild == null)

{

MessageBox.Show("Не открыто ни одной формы!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

return;

}

Table table = (Table)ActiveMdiChild;

var currentRow = table.DataGridView.CurrentRow;

Exam t = (Exam)currentRow?.DataBoundItem;

if (t != null)

{

table.exams.Remove(t);

}

else

{

MessageBox.Show("Не выбран студент!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

}

}

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

{

if (ActiveMdiChild == null)

{

MessageBox.Show("Не открыто ни одной формы", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

return;

}

Table table = (Table)ActiveMdiChild;

Exam t = (Exam)table.DataGridView.CurrentRow?.DataBoundItem;

if (t != null)

{

int index = table.exams.IndexOf(t);

using (EditForm editForm = new EditForm(t))

{

var result = editForm.ShowDialog();

if (result == DialogResult.OK)

{

table.exams[index] = editForm.Result;

}

}

}

else

{

MessageBox.Show("Не выбран студент!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

}

}

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

{

MessageBox.Show("Лабораторная работа №4 \nGitHub: https://github.com/ilyadzh", "О программе", MessageBoxButtons.OK, MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1);

}

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

{

Table table = new Table();

windowNumber++;

table.MdiParent = this;

table.Text = "Таблица №" + windowNumber;

table.Show();

}

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

{

OpenFileDialog dialog = new OpenFileDialog();

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

{

Exam[] fields;

try

{

BinaryFormatter formatter = new BinaryFormatter();

FileStream fs = File.Open(dialog.FileName, FileMode.OpenOrCreate);

fields = (Exam[])formatter.Deserialize(fs);

}

catch (IOException)

{

return;

}

catch (InvalidCastException)

{

MessageBox.Show("Неверный тип файла", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

catch (SerializationException)

{

MessageBox.Show("Неверный тип файла", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

Table table = new Table(fields);

windowNumber++;

table.MdiParent = this;

table.Text = "Таблица №" + windowNumber;

table.Show();

}

}

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

{

if (ActiveMdiChild == null)

{

MessageBox.Show("Не открыто ни одной таблицы!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

return;

}

SaveFileDialog dialog = new SaveFileDialog();

Table table = (Table)ActiveMdiChild;

dialog.Filter = "dat-файл(\*.dat)|\*.dat";

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

{

BinaryFormatter formatter = new BinaryFormatter();

FileStream fs = File.Open(dialog.FileName, FileMode.OpenOrCreate);

formatter.Serialize(fs, table.exams.ToArray());

fs.Close();

}

}

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

{

if (ActiveMdiChild == null)

{

MessageBox.Show("Не открыто ни одной формы!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);

return;

}

SelectForm selectForm = new SelectForm((Table)ActiveMdiChild);

DialogResult result = selectForm.ShowDialog();

if (result == DialogResult.OK)

{

Table tableSelect = new Table();

tableSelect.MdiParent = this;

tableSelect.Text = "Выборка";

tableSelect.Show();

tableSelect.DataGridView.DataSource = selectForm.results;

}

}

}

}

Файл Exam.cs

using System;

using System.Drawing;

namespace WindowsFormsApp2

{

[Serializable]

public class Exam

{

public string FirstName { get; set; }

public string LastName { get; set; }

public DateTime Date { get; set; }

public int Mark { get; set; }

public string Questions { get; set; }

public string Photo { get; set; }

public Exam(string FirstName, string LastName, DateTime Date, int Mark, string Questions, string Photo)

{

this.FirstName = FirstName;

this.LastName = LastName;

this.Date = Date;

this.Mark = Mark;

this.Questions = Questions;

this.Photo = Photo;

}

}

}

4 РЕЗУЛЬТАТЫ

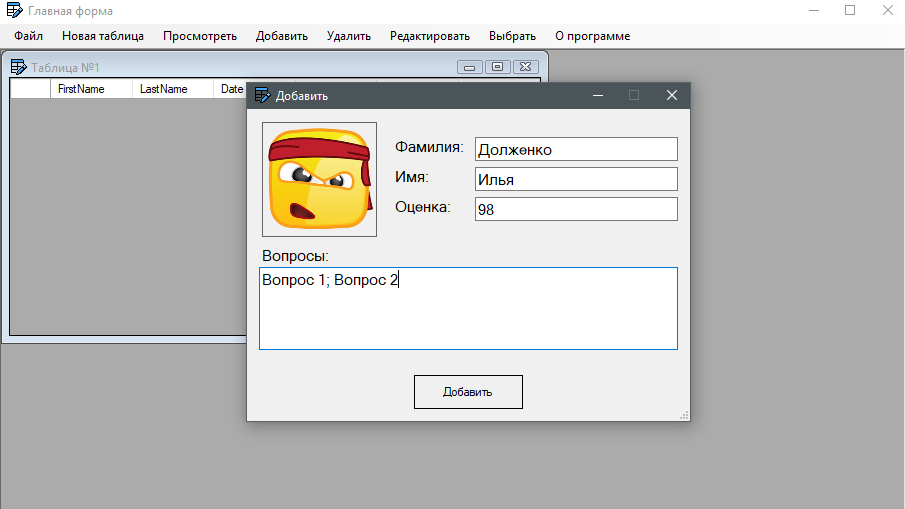


Рисунок 1 – Процесс добавления студента в таблицу

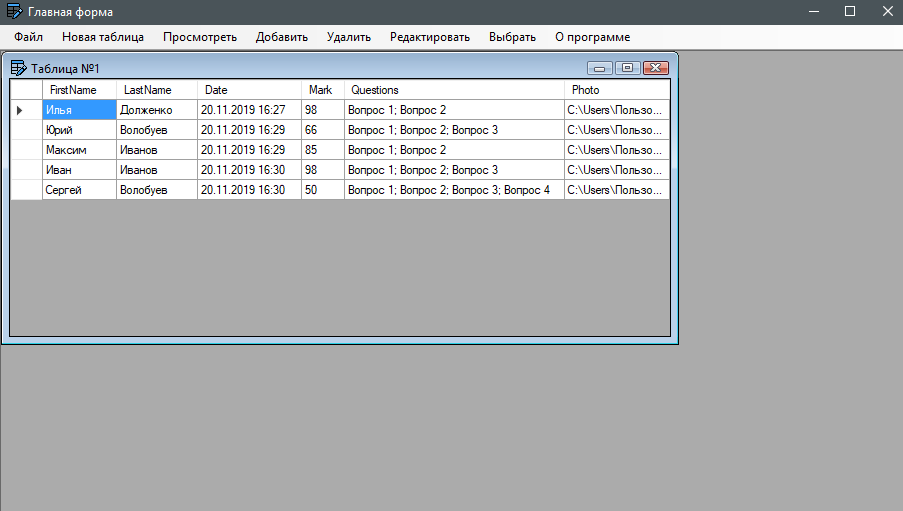


Рисунок 2 – Результат добавления нескольких студентов в таблицу

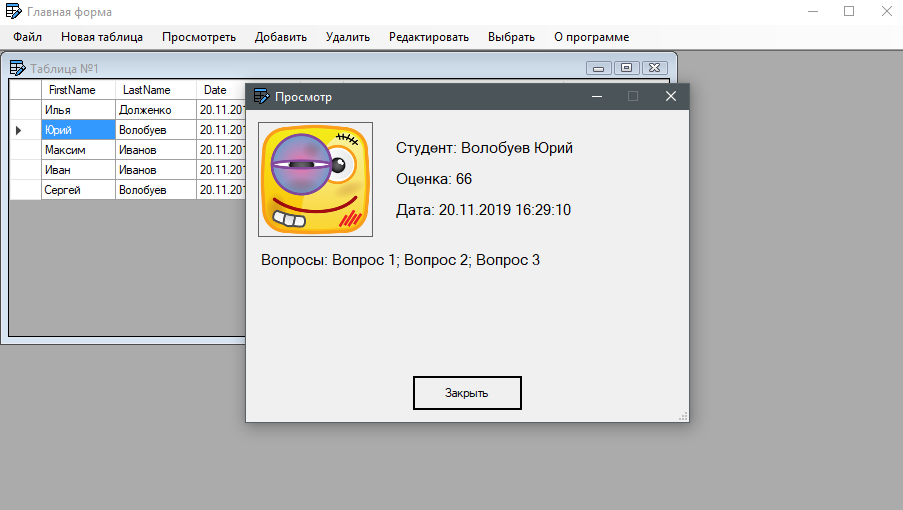


Рисунок 3 – Просмотр студента

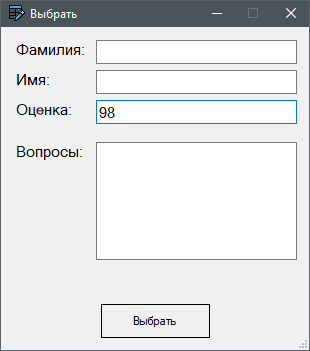


Рисунок 4 – Параметры выборки

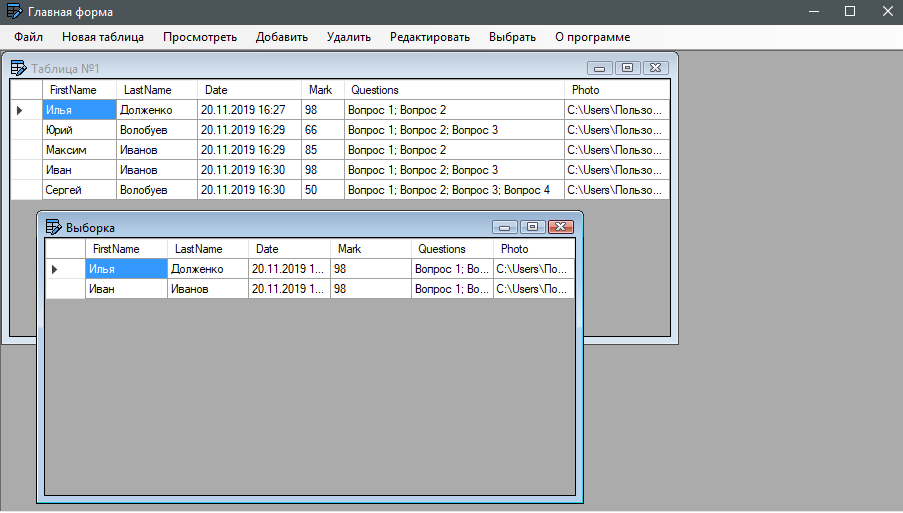


Рисунок 5 – Результат выборки

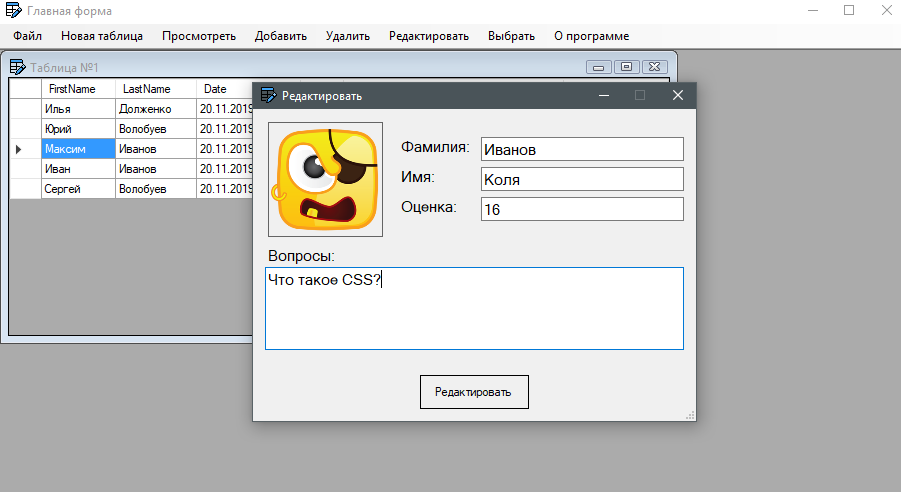


Рисунок 6 – Процесс редактирования студента

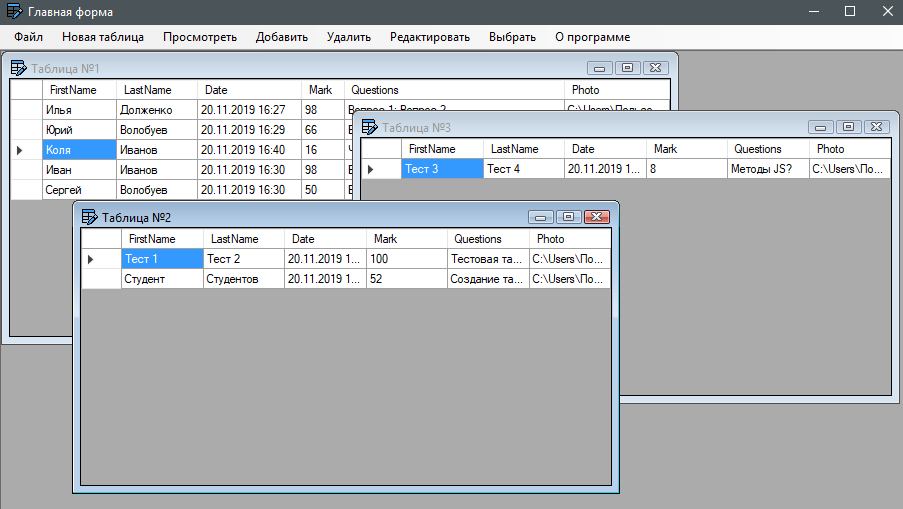


Рисунок 7 – Возможность работы с несколькими таблицами

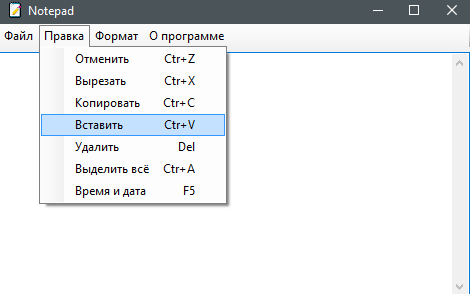


Рисунок 8 – Блокнот

ВЫВОДЫ

В процессе выполнения лабораторной работы были изучены особенности разработки MDI-приложений в Visual Studio. Изучены способы сохранения данных в файл и загрузки из файла. Освоены механизм сериализации и десериализации объектов.

Была написан текстовый редактор Notepad, реализующий функции создания, открытия и сохранения текста. Для работы с текстом реализованы возможности копирования, вставки, удаления, выделения всего текста и вставки текущего времени. Имеется возможность изменить цвет поля и стиль текста.

Была дописана программа, работающая со списком студентов. В таблицу добавлено новое поле “Фото студента”, появилась возможность добавления нескольких таблиц, сохранения их в файл и новое окно с параметрами выборки из данных таблиц.