**Занятие № 21**

**Дата выполнения работы:** 15.05.2023

# **Тема работы:** «Технология Windows Form. Работа с формами»

**Ход работы**

**Задание 1**

Создать форму (или формы для ввода агрегируемых объектов).

Разместить на ней ЭУ для ввода/вывода информации об объекте (создать свои

типы). На форме разместить не менее 9-и различных типов ЭУ (радиокнопки,

списки, поля ввода, метки, кнопки, слайдеры, календарь и т.д.).

Создать дополнительные кнопки для сохранения введенной

информации и вывода (отображения сохраненных данных). Запись

сохраняемых объектов и чтение выполнять в./из файл типа xml и json.

Выполнить валидацию вводимых пользователем данных.

Университет: Объект – «Студент». Поля: ФИО, возраст,

специальность, дата рождения, курс, группа, средний балл, пол,

адреса и др. Агрегируемый объект – «Адрес». Поля: город, индекс,

улица, дом, квартира.

Дополнительно: Агрегируемый объект – «Место текущей

работы». Поля: компания, должность, страж и т.д.

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml;

using System.Xml.Serialization;

using Newtonsoft.Json;

namespace App

{

public partial class Form1 : Form

{

private List<Student> students = new List<Student>();

public Form1()

{

InitializeComponent();

}

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

{

cmbSpecialty.Items.AddRange(new string[] { "Математика", "Физика", "Химия", "Информатика" });

}

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

{

// Валидация вводимых данных

if (string.IsNullOrWhiteSpace(txtName.Text))

{

MessageBox.Show("Введите ФИО студента");

return;

}

if (string.IsNullOrWhiteSpace(cmbSpecialty.Text))

{

MessageBox.Show("Выберите специальность");

return;

}

// Создание и добавление студента

var student = new Student()

{

Name = txtName.Text,

Age = (int)numAge.Value,

Specialty = cmbSpecialty.Text,

BirthDate = dateBirthDate.Value,

Course = (int)numCourse.Value,

Group = txtGroup.Text,

GPA = (double)numGPA.Value,

Gender = radMale.Checked ? "Мужской" : "Женский",

Address = new Address()

{

City = txtCity.Text,

PostalCode = txtPostalCode.Text,

Street = txtStreet.Text,

HouseNumber = txtHouseNumber.Text,

ApartmentNumber = txtApartmentNumber.Text

}

};

students.Add(student);

ClearForm();

}

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

{

// Отображение сохраненных данных

var output = "";

foreach (var student in students)

{

output += "ФИО: " + student.Name + Environment.NewLine;

output += "Возраст: " + student.Age + Environment.NewLine;

output += "Специальность: " + student.Specialty + Environment.NewLine;

output += "Дата рождения: " + student.BirthDate.ToShortDateString() + Environment.NewLine;

output += "Курс: " + student.Course + Environment.NewLine;

output += "Группа: " + student.Group + Environment.NewLine;

output += "Средний балл: " + student.GPA + Environment.NewLine;

output += "Пол: " + student.Gender + Environment.NewLine;

output += "Адрес: " + Environment.NewLine;

output += "Город: " + student.Address.City + Environment.NewLine;

output += "Индекс: " + student.Address.PostalCode + Environment.NewLine;

output += "Улица: " + student.Address.Street + Environment.NewLine;

output += "Дом: " + student.Address.HouseNumber + Environment.NewLine;

output += "Квартира: " + student.Address.ApartmentNumber + Environment.NewLine;

output += Environment.NewLine;

}

MessageBox.Show(output);

}

private void ClearForm()

{

// Очистка полей формы

txtName.Text = "";

numAge.Value = 0;

cmbSpecialty.SelectedIndex = -1;

dateBirthDate.Value = DateTime.Now;

numCourse.Value = 1;

txtGroup.Text = "";

numGPA.Value = 0;

radMale.Checked = true;

txtCity.Text = "";

txtPostalCode.Text = "";

txtStreet.Text = "";

txtHouseNumber.Text = "";

txtApartmentNumber.Text = "";

}

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

{

// Сохранение данных в XML и JSON файлы

using (var xmlWriter = XmlWriter.Create("students.xml"))

{

var serializer = new XmlSerializer(typeof(List<Student>));

serializer.Serialize(xmlWriter, students);

}

var json = JsonConvert.SerializeObject(students);

System.IO.File.WriteAllText("students.json", json);

MessageBox.Show("Данные сохранены в файлы XML и JSON");

}

}

public class Student

{

public string Name { get; set; }

public int Age { get; set; }

public string Specialty { get; set; }

public DateTime BirthDate { get; set; }

public int Course { get; set; }

public string Group { get; set; }

public double GPA { get; set; }

public string Gender { get; set; }

public Address Address { get; set; }

}

public class Address

{

public string City { get; set; }

public string PostalCode { get; set; }

public string Street { get; set; }

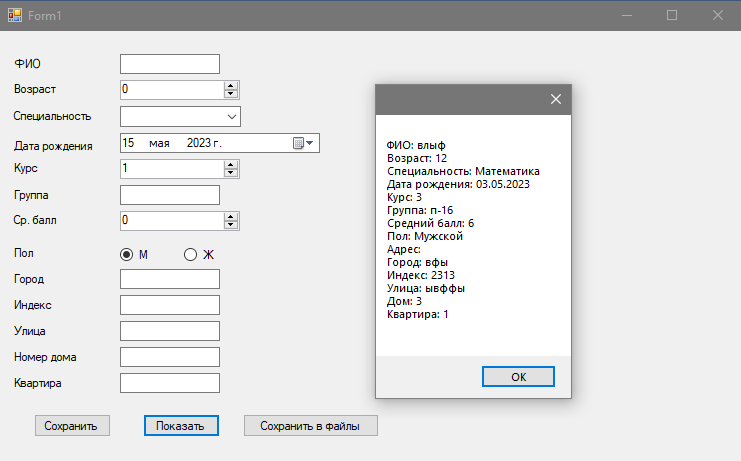
public string HouseNumber { get; set; }

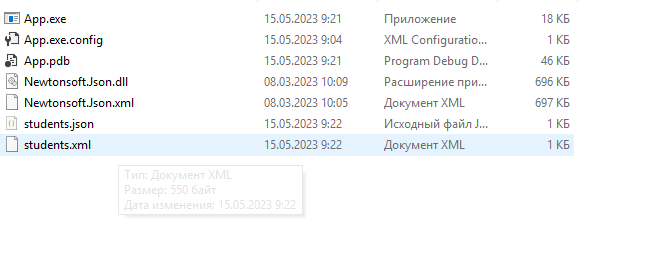
public string ApartmentNumber { get; set; }

}

}

**Результат:**

****

****

**Вывод:** Технология Windows Forms (WinForms) позволяет создавать графические пользовательские интерфейсы для приложений на Windows. Работа с формами включает размещение элементов управления, обработку событий, валидацию данных и сохранение информации. WinForms обеспечивает удобный способ создания интерфейсов для Windows-приложений.