Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М.Машерова

**Отчет**

по учебной практике по программированию

«Модуль 6. Базы данных»

Выполнил учащийся Казаченко Н.С.

группы 3ПОИС23 06.10.2025 г.

Проверил Алейников М.А.

06.10.2025 г.

Орша, 2025

**Цель:** закрепить знания о базах данных на языке программирования С#.

**Модуль 6 Базы данных (1 вариант):**

Создайте базу данных для учета сотрудников в организации. Реализуйте приложение для добавления, редактирования и удаления записей о сотрудниках.

**Задание 1.**

Для выполнения данного задания требовалось написать программу, которая будет с использованием базы данных работать с сотрудниками предприятия.

Код программы представлен в листинге 1.

Запущенное окно программы представлено на рисунке 1.

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.IO;

namespace \_6.\_1

{

public partial class Form1 : Form

{

private DataTable employeesTable;

private readonly string dataFilePath = Path.Combine(Application.StartupPath, "employees.xml");

public Form1()

{

InitializeComponent();

InitializeDataModel();

}

private void InitializeDataModel()

{

employeesTable = new DataTable("Employees");

employeesTable.Columns.Add("Id", typeof(int));

employeesTable.Columns.Add("Name", typeof(string));

employeesTable.Columns.Add("Position", typeof(string));

employeesTable.Columns.Add("Department", typeof(string));

employeesTable.Columns.Add("BirthDate", typeof(DateTime));

employeesTable.Columns.Add("Salary", typeof(decimal));

employeesTable.PrimaryKey = new DataColumn[] { employeesTable.Columns["Id"] };

}

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

{

LoadData();

BindGrid();

}

private void BindGrid()

{

dgvEmployees.DataSource = employeesTable;

if (dgvEmployees.Columns.Contains("Id")) dgvEmployees.Columns["Id"].Visible = false;

if (dgvEmployees.Columns.Contains("Name")) dgvEmployees.Columns["Name"].HeaderText = "ФИО";

if (dgvEmployees.Columns.Contains("Position")) dgvEmployees.Columns["Position"].HeaderText = "Должность";

if (dgvEmployees.Columns.Contains("Department")) dgvEmployees.Columns["Department"].HeaderText = "Отдел";

if (dgvEmployees.Columns.Contains("BirthDate")) dgvEmployees.Columns["BirthDate"].HeaderText = "Дата рождения";

if (dgvEmployees.Columns.Contains("Salary")) dgvEmployees.Columns["Salary"].HeaderText = "Зарплата";

dgvEmployees.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

}

private void LoadData()

{

if (File.Exists(dataFilePath))

{

DataSet ds = new DataSet();

ds.ReadXml(dataFilePath);

if (ds.Tables.Contains("Employees"))

{

employeesTable = ds.Tables["Employees"];

if (!employeesTable.Columns.Contains("Id")) employeesTable.Columns.Add("Id", typeof(int));

employeesTable.PrimaryKey = new DataColumn[] { employeesTable.Columns["Id"] };

}

}

}

private void SaveData()

{

DataSet ds = new DataSet();

ds.Tables.Add(employeesTable.Copy());

ds.WriteXml(dataFilePath, XmlWriteMode.WriteSchema);

}

private int GenerateNextId()

{

if (employeesTable.Rows.Count == 0) return 1;

return employeesTable.AsEnumerable().Max(r => r.Field<int>("Id")) + 1;

}

private void ClearInputs()

{

txtName.Text = string.Empty;

txtPosition.Text = string.Empty;

txtDepartment.Text = string.Empty;

dtpBirthDate.Value = DateTime.Today;

numSalary.Value = 0;

}

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

{

if (string.IsNullOrWhiteSpace(txtName.Text))

{

MessageBox.Show("Введите ФИО", "Валидация", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

DataRow row = employeesTable.NewRow();

row["Id"] = GenerateNextId();

row["Name"] = txtName.Text.Trim();

row["Position"] = txtPosition.Text.Trim();

row["Department"] = txtDepartment.Text.Trim();

row["BirthDate"] = dtpBirthDate.Value.Date;

row["Salary"] = numSalary.Value;

employeesTable.Rows.Add(row);

ClearInputs();

}

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

{

if (dgvEmployees.CurrentRow == null) return;

DataRowView drv = dgvEmployees.CurrentRow.DataBoundItem as DataRowView;

if (drv == null) return;

if (string.IsNullOrWhiteSpace(txtName.Text))

{

MessageBox.Show("Введите ФИО", "Валидация", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

drv["Name"] = txtName.Text.Trim();

drv["Position"] = txtPosition.Text.Trim();

drv["Department"] = txtDepartment.Text.Trim();

drv["BirthDate"] = dtpBirthDate.Value.Date;

drv["Salary"] = numSalary.Value;

}

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

{

if (dgvEmployees.CurrentRow == null) return;

DataRowView drv = dgvEmployees.CurrentRow.DataBoundItem as DataRowView;

if (drv == null) return;

if (MessageBox.Show("Удалить выбранную запись?", "Подтверждение", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

drv.Row.Delete();

ClearInputs();

}

}

private void dgvEmployees\_SelectionChanged(object sender, EventArgs e)

{

if (dgvEmployees.CurrentRow == null) return;

DataRowView drv = dgvEmployees.CurrentRow.DataBoundItem as DataRowView;

if (drv == null) return;

txtName.Text = Convert.ToString(drv["Name"]);

txtPosition.Text = Convert.ToString(drv["Position"]);

txtDepartment.Text = Convert.ToString(drv["Department"]);

DateTime bd;

if (DateTime.TryParse(Convert.ToString(drv["BirthDate"]), out bd))

dtpBirthDate.Value = bd;

decimal sal;

if (decimal.TryParse(Convert.ToString(drv["Salary"]), out sal))

numSalary.Value = sal;

}

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

{

try

{

SaveData();

MessageBox.Show("Данные сохранены", "Готово", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка сохранения: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

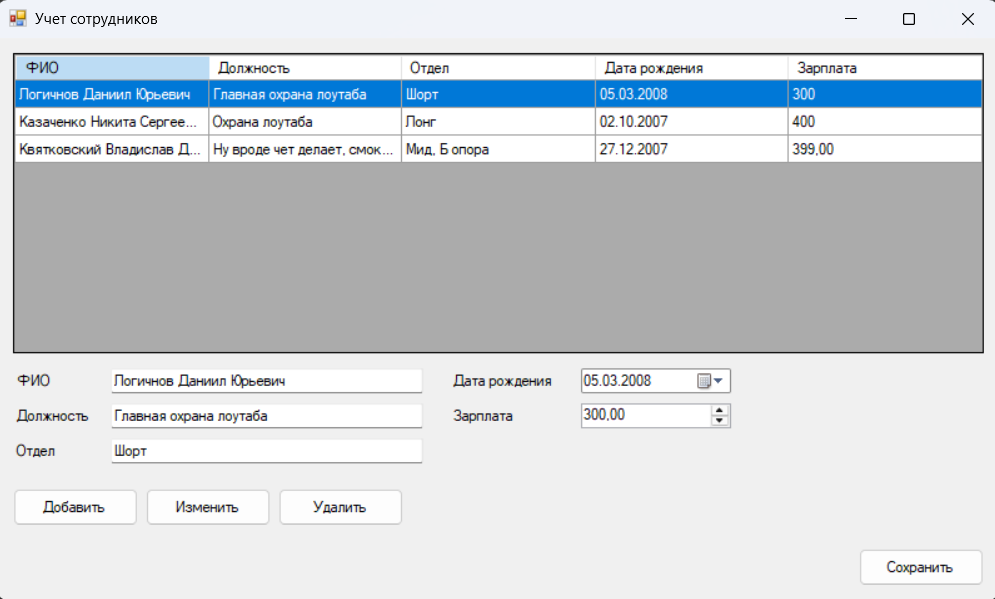


Рисунок 1 – Программа учета сотрудников

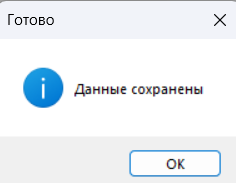


Рисунок 2 – Подтверждение сохранения данных в бд

**Ссылка на репозиторий:**

[**https://github.com/Seyko00/PraktikaPoProgrammirovaniu**](https://github.com/Seyko00/PraktikaPoProgrammirovaniu)

**Вывод:** в ходе выполнения модуля 6 были закреплены знания о базах данных на ЯП С#