**Занятие № 7**

**Дата выполнения работы:** 26.04.2023

# **Тема работы:** «Перечисления. Структуры.»

**Ход работы**

**Задание 1**

В соответствии со своим вариантом разработать программу для работы с

таблицей данных. Определить структурированный тип, меню для работы с

массивом структур.

В перечень обязательных должны войти функции:

- ввод элементов структуры с клавиатуры;

- вывод элементов структуры;

- удаление всех структурированных переменных;

- удаление заданной структурированной переменной;

- поиск информации.

1. Горожанин. Ф.И.О., дата рождения, адрес, пол (м, ж). Реализовать выборку по полу и году рождения.

**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;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

// Структура "Citizen"

struct Citizen

{

public string FullName;

public DateTime Birthdate;

public string Address;

public char Gender;

}

private List<Citizen> citizensList = new List<Citizen>(); // Список горожан

private void AddCitizen(string fullName, DateTime birthdate, string address, char gender)

{

Citizen citizen = new Citizen

{

FullName = fullName,

Birthdate = birthdate,

Address = address,

Gender = gender

};

citizensList.Add(citizen);

}

// Функция для очистки списка горожан

private void ClearCitizensList()

{

citizensList.Clear();

}

// Функция для поиска горожанина по полу и году рождения

private List<Citizen> SearchCitizens(char gender, int birthYear)

{

return citizensList.Where(c => c.Gender == gender && c.Birthdate.Year == birthYear).ToList();

}

private void InitializeNumericUpDown()

{

nudSearchBirthYear = new System.Windows.Forms.NumericUpDown();

nudSearchBirthYear.Location = new System.Drawing.Point(150, 50); // Установка координат расположения

nudSearchBirthYear.Size = new System.Drawing.Size(100, 20); // Установка размеров

nudSearchBirthYear.Minimum = 1900; // Минимальное значение

nudSearchBirthYear.Maximum = DateTime.Now.Year; // Максимальное значение - текущий год

this.Controls.Add(nudSearchBirthYear); // Добавление компонента на форму

}

// Функция для удаления горожанина из списка по индексу

private void RemoveCitizen(int index)

{

if (index >= 0 && index < citizensList.Count)

{

citizensList.RemoveAt(index);

}

}

public Form1()

{

InitializeComponent();

}

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

{

string fullName = tbFullName.Text;

DateTime birthdate = dtpBirthdate.Value;

string address = tbAddress.Text;

char gender = rbMale.Checked ? 'М' : 'Ж';

AddCitizen(fullName, birthdate, address, gender);

UpdateDataGridView();

ClearInputFields();

}

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

{

int selectedIndex = dgvCitizensList.SelectedCells[0].RowIndex;

RemoveCitizen(selectedIndex);

UpdateDataGridView();

}

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

{

char gender = rbSearchMale.Checked ? 'М' : 'Ж';

int birthYear = (int)nudSearchBirthYear.Value;

List<Citizen> searchResult = SearchCitizens(gender, birthYear);

UpdateDataGridView(searchResult);

}

private void UpdateDataGridView(List<Citizen> dataSource = null)

{

dgvCitizensList.Rows.Clear();

List<Citizen> data = dataSource ?? citizensList;

foreach (var citizen in data)

{

dgvCitizensList.Rows.Add(citizen.FullName, citizen.Birthdate.ToShortDateString(), citizen.Address, citizen.Gender);

}

}

// Функция для очистки полей ввода

private void ClearInputFields()

{

tbFullName.Clear();

dtpBirthdate.Value = DateTime.Now;

tbAddress.Clear();

rbMale.Checked = true;

}

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

{

// Устанавливаем формат даты для DateTimePicker

dtpBirthdate.Format = DateTimePickerFormat.Short;

// Добавление столбца "ФИО"

DataGridViewTextBoxColumn colFullName = new DataGridViewTextBoxColumn();

colFullName.Name = "colFullName";

colFullName.HeaderText = "ФИО";

dgvCitizensList.Columns.Add(colFullName);

// Добавление столбца "Дата рождения"

DataGridViewTextBoxColumn colBirthdate = new DataGridViewTextBoxColumn();

colBirthdate.Name = "colBirthdate";

colBirthdate.HeaderText = "Дата рождения";

dgvCitizensList.Columns.Add(colBirthdate);

// Добавление столбца "Адрес"

DataGridViewTextBoxColumn colAddress = new DataGridViewTextBoxColumn();

colAddress.Name = "colAddress";

colAddress.HeaderText = "Адрес";

dgvCitizensList.Columns.Add(colAddress);

// Добавление столбца "Пол"

DataGridViewTextBoxColumn colGender = new DataGridViewTextBoxColumn();

colGender.Name = "colGender";

colGender.HeaderText = "Пол";

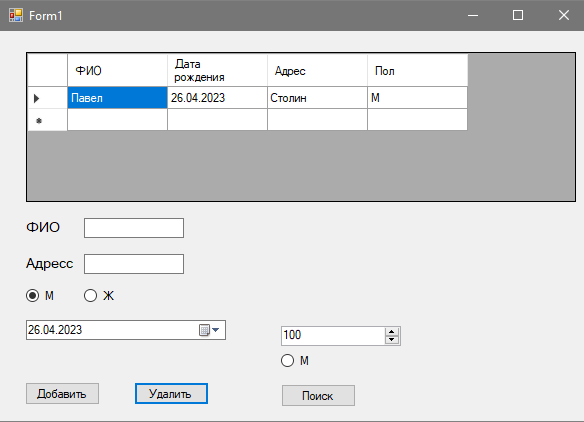
dgvCitizensList.Columns.Add(colGender);

}

}

}

**Результат:**



**Вывод:** после изучения перечислений и структур в C# Windows Forms я научился использовать эти мощные средства языка программирования для более удобной и эффективной работы с данными в моих приложениях.

Перечисления позволили мне создавать пользовательские типы данных с ограниченным набором значений, что особенно полезно для представления наборов констант или опций выбора. Я могу объявлять перечисления с помощью ключевого слова "enum" и определять список значений, которые могут принимать эти перечисления. Затем я могу использовать эти перечисления в моих структурах и классах, чтобы описывать состояние объектов и их свойства.

Благодаря изучению перечислений и структур в C# Windows Forms, я теперь имею мощные инструменты для более удобной и гибкой работы с данными в своих приложениях, что позволяет мне создавать более эффективные и функциональные программы.