**Занятие № 12**

**Дата выполнения работы:** 03.05.2023

# **Тема работы:** «Разработка, отладка и испытание программ на фильтрацию выборки»

**Ход работы**

**Задание 1**

Создать два класса по темам, представленным ниже. Осуществить выбор-

ку на основе двух источников данных с использованием LINQ.

1. Больница.

**Листинг программы:**

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;

namespace App

{

public partial class Form1 : Form

{

private Hospital hospital;

public Form1()

{

InitializeComponent();

hospital = new Hospital();

// Добавляем пациентов в больницу

hospital.AddPatient(new Patient { Id = 1, Name = "Иванов Иван Иванович", DateOfBirth = new DateTime(1990, 5, 1), Diagnosis = "Грипп", IsAdmitted = true });

hospital.AddPatient(new Patient { Id = 2, Name = "Петров Петр Петрович", DateOfBirth = new DateTime(1980, 12, 15), Diagnosis = "Ангина", IsAdmitted = true });

hospital.AddPatient(new Patient { Id = 3, Name = "Сидоров Сидор Сидорович", DateOfBirth = new DateTime(1975, 8, 23), Diagnosis = "Простуда", IsAdmitted = false });

hospital.AddPatient(new Patient { Id = 4, Name = "Кузнецова Екатерина Владимировна", DateOfBirth = new DateTime(1995, 3, 10), Diagnosis = "Перелом", IsAdmitted = false });

}

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

{

var patients = hospital.GetPatients().Where(p => p.IsAdmitted);

listBox1.Items.Clear();

foreach (var patient in patients)

{

listBox1.Items.Add($"{patient.Name} ({patient.DateOfBirth.ToShortDateString()})");

}

}

}

public class Patient

{

public int Id { get; set; }

public string Name { get; set; }

public DateTime DateOfBirth { get; set; }

public string Diagnosis { get; set; }

public bool IsAdmitted { get; set; }

}

public class Hospital

{

private List<Patient> patients;

public Hospital()

{

patients = new List<Patient>();

}

public void AddPatient(Patient patient)

{

patients.Add(patient);

}

public void RemovePatient(Patient patient)

{

patients.Remove(patient);

}

public List<Patient> GetPatients()

{

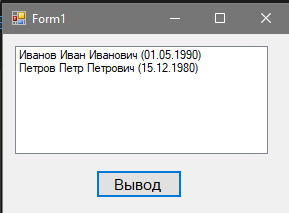
return patients;

}

}

}

**Результат:**

****

**Вывод:** Разработка, отладка и испытание программ на фильтрацию выборки - важная задача при работе с данными в программировании. Она позволяет получать нужную информацию из большого объема данных с помощью фильтров и сортировок, что экономит время и повышает эффективность работы программы. В качестве инструментов для фильтрации данных часто используются LINQ-запросы, которые позволяют выбирать нужные элементы из коллекций. При разработке программ на фильтрацию выборки необходимо учитывать возможные ошибки и баги, поэтому отладка и испытание программ являются неотъемлемой частью этого процесса. В целом, правильное использование фильтрации выборки позволяет улучшить качество и скорость работы программы, что является важным фактором в различных областях программирования.