**Занятие № 12**

**Дата выполнения работы:** 03.05.2023

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

**Ход работы**

**Меню:**

**Задание 1**

**Создать два класса по теме, представленной ниже. Осуществить выбор-ку на основе двух источников данных с использованием LINQ.**

**Информационные технологии.**

**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 \_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

class Person

{

public string Name { get; set; }

public int Age { get; set; }

public string Email { get; set; }

public string[] Interests { get; set; }

}

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

{

List<Person> people = new List<Person>

{

new Person { Name = "John", Age = 25, Email = "john@example.com", Interests = new string[] { "Coding", "Gaming", "Music" } },

new Person { Name = "Emily", Age = 30, Email = "emily@example.com", Interests = new string[] { "Hiking", "Photography", "Cooking" } },

new Person { Name = "Michael", Age = 20, Email = "michael@example.com", Interests = new string[] { "Sports", "Traveling", "Reading" } }

};

var coders = people.Where(p => p.Interests.Contains("Coding"));

var result = string.Join(Environment.NewLine, coders.Select(p => $"{p.Name} ({p.Age})"));

txtResult.Text = result;

}

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

{

List<Person> people = new List<Person>

{

new Person { Name = "John", Age = 25, Email = "john@example.com", Interests = new string[] { "Coding", "Gaming", "Music" } },

new Person { Name = "Emily", Age = 30, Email = "emily@example.com", Interests = new string[] { "Hiking", "Photography", "Cooking" } },

new Person { Name = "Michael", Age = 20, Email = "michael@example.com", Interests = new string[] { "Sports", "Traveling", "Reading" } }

};

var sortedPeople = people.OrderBy(p => p.Name);

var result = string.Join(Environment.NewLine, sortedPeople.Select(p => $"{p.Name} ({p.Age})"));

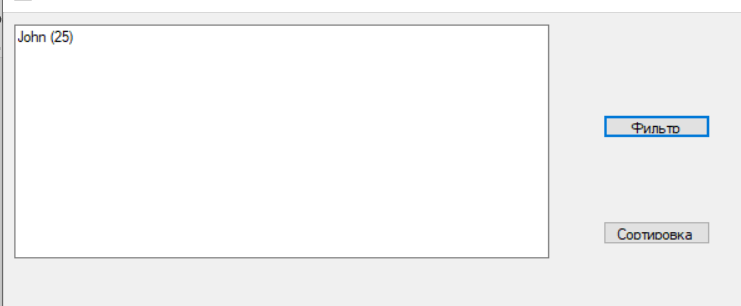
txtResult.Text = result;

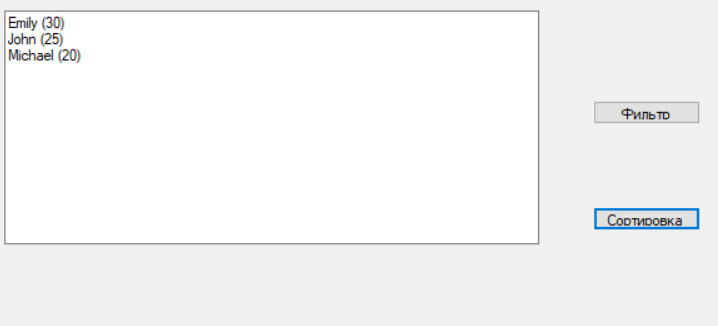
}

}

}

**Результат:**





**Вывод:**Разработка и отладка относятся к различным этапам процесса создания программного обеспечения.

Разработка - это процесс создания программного обеспечения, который включает в себя анализ требований, проектирование архитектуры, написание кода, тестирование и внесение изменений в продукт.

Отладка - это процесс выявления и исправления ошибок в программном обеспечении. Этот процесс начинается после того, как программа была написана и протестирована, и обнаружены ошибки, которые могут привести к неправильной работе приложения.

Таким образом, разработка и отладка - это два разных этапа в жизненном цикле разработки программного обеспечения. Разработка предшествует отладке и заключается в создании программного продукта, в то время как отладка осуществляется после завершения разработки, чтобы исправить ошибки и гарантировать корректную работу программного продукта.