

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационной безопасности

ОТЧЕТ
по лабораторной работе №11
по дисциплине «Объектно-ориентированное программирование»
Тема: Использование LINQ

Студент гр. 3363

Преподаватель

Овсейчик Н. И.,
Минко Д. А.,
Гончаренко О. Д.

Новакова Н. Е.

Санкт-Петербург
2024

Цель работы

Цель лабораторной работы – освоить использование технологий LINQ в программировании на платформе .NET путем создания приложения Windows Forms, работы с методами сортировки, фильтрации и преобразования коллекций данных. В процессе выполнения необходимо реализовать функциональность вывода списка исполнителей и их альбомов с использованием методов LINQ. Также требуется изучить обработку событий формы для взаимодействия с элементами управления и организовать сортировку данных по алфавиту и дате.

Анализ задачи

Необходимо:

1. Реализовать программу, которая из переданного списка альбомов выводит в графический интерфейс уникальный список исполнителей, представленных в этом списке.
2. Модифицировать программу так, чтобы при выборе исполнителя отображались только те альбомы, которые принадлежат выбранному исполнителю.
3. Доработать функционал программы, добавив сортировку: исполнителей – в алфавитном порядке, а альбомов – по дате выхода, начиная с самых новых.

ХОД РАБОТЫ

Упражнение №1

1. Создание проекта и изучение классов

Microsoft Visual Studio 2020 был создан новый проект типа *Windows Forms Application* с названием *OopLabs.Linq*. В проект был добавлен файл *Artist.cs*. Изучено описание класса *Album*, который включает поля для хранения информации о музыкальных альбомах: имя исполнителя, название альбома и дата выхода. Также в классе *Album* реализован статический метод *Album.GetAlbums()*, возвращающий список альбомов нескольких групп.

2. Реализация вывода списка исполнителей

На главной форме программы был добавлен текстовый элемент управления *Label* с надписью «Исполнители» и элемент *ListBox*, предназначенный для отображения списка исполнителей.

В методе *MainForm_Load* был реализован код, который получает данные из метода *Album.GetAlbums()*, применяя LINQ-функции *Select()*, *Distinct()* и *ToList()* для извлечения уникальных имен исполнителей. Программа была скомпилирована и запущена, результат корректно отображал список исполнителей (рис. 1).

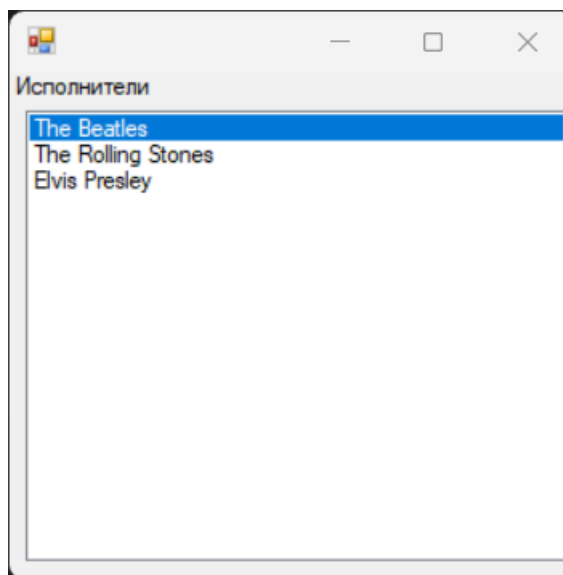


Рисунок 1 – Отображение списка уникальных исполнителей

Упражнение №2

3. Реализация вывода альбомов выбранного исполнителя

На форме был добавлен текстовый элемент управления Label с надписью «Альбомы» и дополнительный элемент ListBox для отображения списка альбомов.

В обработчике события `SelectedIndexChanged` элемента `ArtistsListBox` был реализован код, фильтрующий список альбомов, основываясь на выбранном исполнителе. Для фильтрации использовалась LINQ-функция `Where()`. После компиляции и запуска программы список альбомов корректно обновлялся при выборе исполнителя из списка (рис. 2-5).

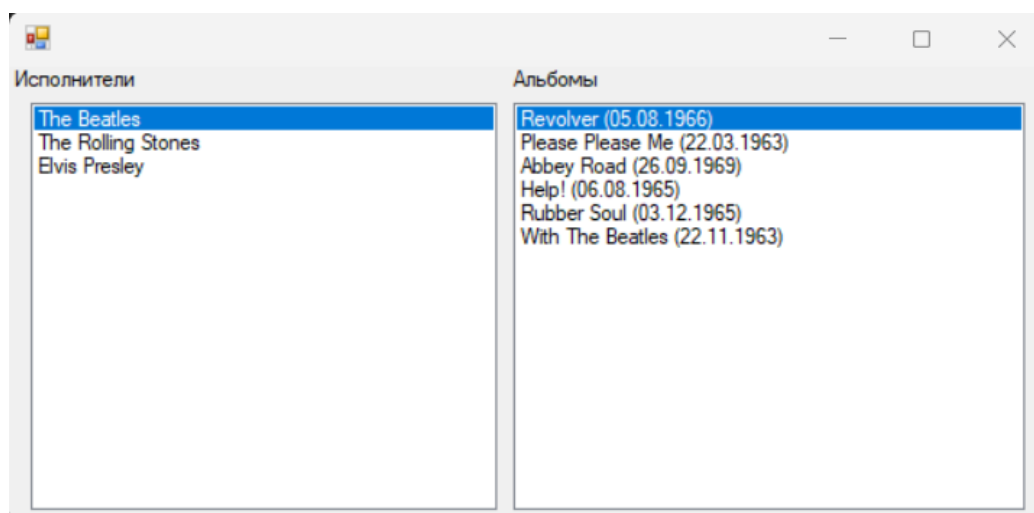


Рисунок 2 – Фильтрация альбомов по исполнителю

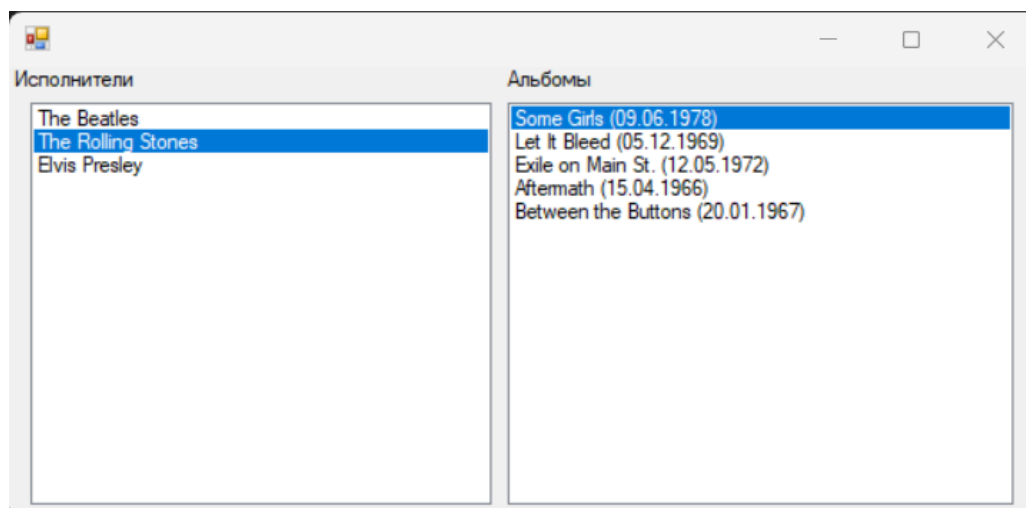


Рисунок 3 – Фильтрация альбомов по исполнителю

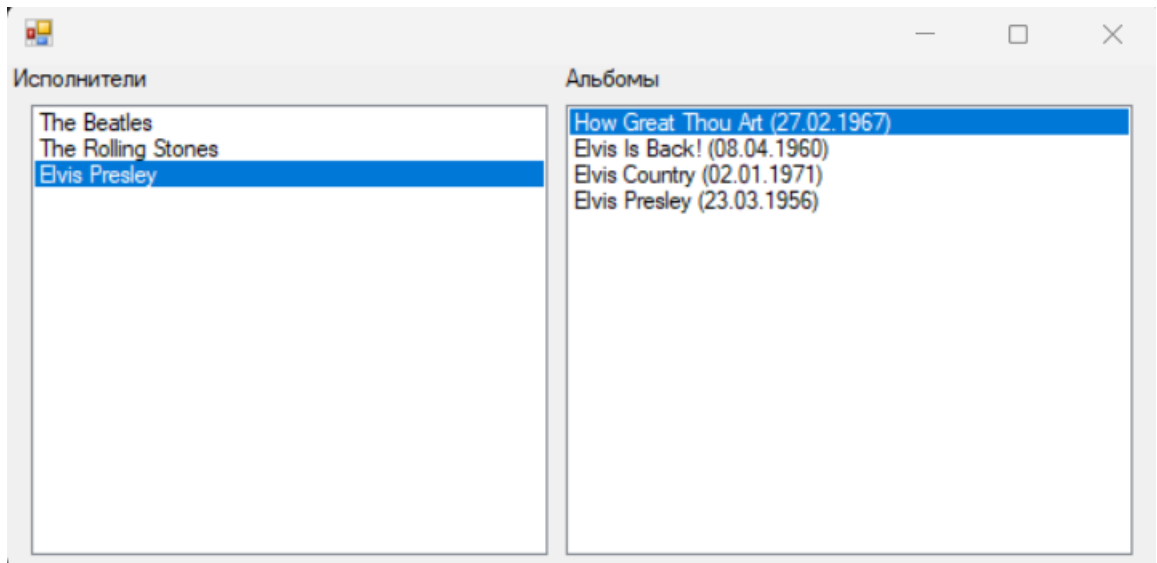


Рисунок 4 – Фильтрация альбомов по исполнителю

Упражнение №3

4. Реализация сортировки данных

Для упорядочивания данных был изменен метод `MainForm_Load`: список исполнителей сортировался в алфавитном порядке с использованием LINQ-функции `OrderBy()`. Также был модифицирован метод `SelectedIndexChanged` для сортировки альбомов по дате выхода в порядке убывания с помощью `OrderByDescending()`. Программа была успешно скомпилирована и запущена, данные корректно отображались в упорядоченном виде (рис. 5-7).

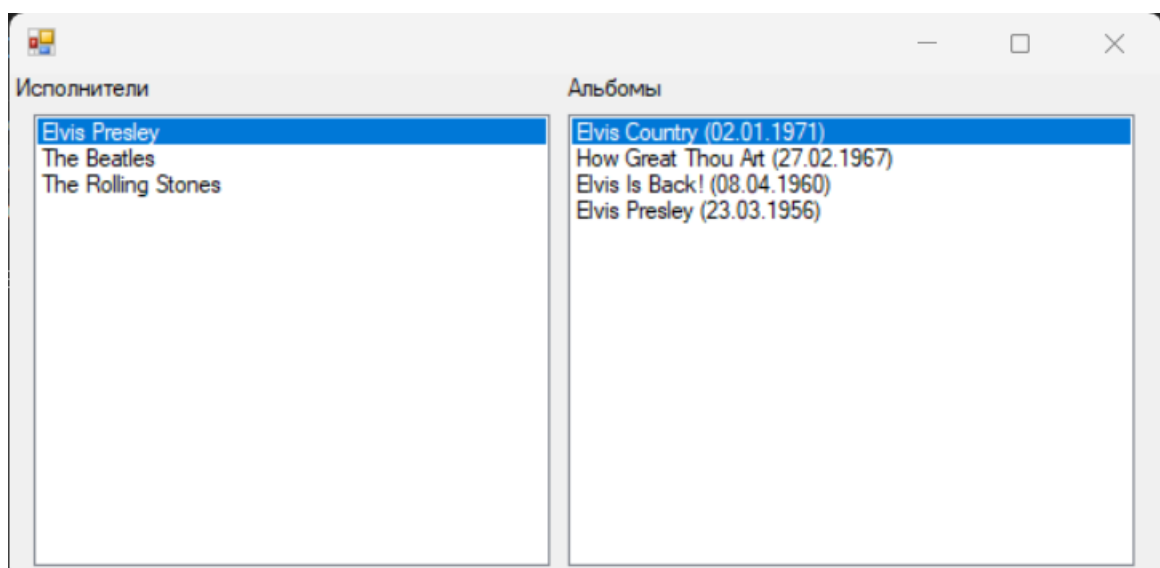


Рисунок 5 – Сортировка списка исполнителей и альбомов

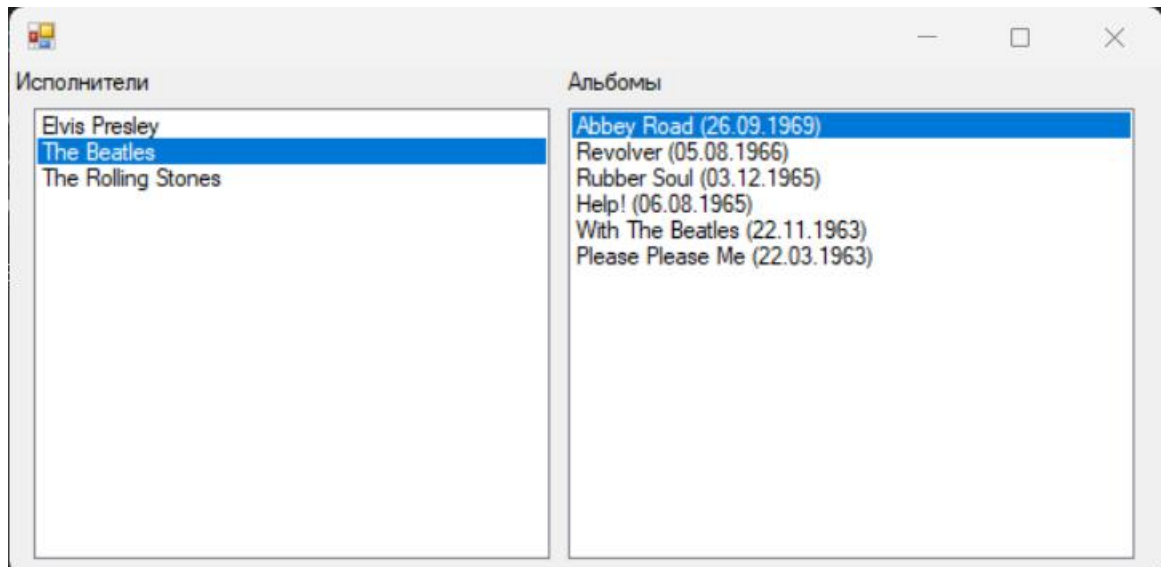


Рисунок 6 – Сортировка списка исполнителей и альбомов

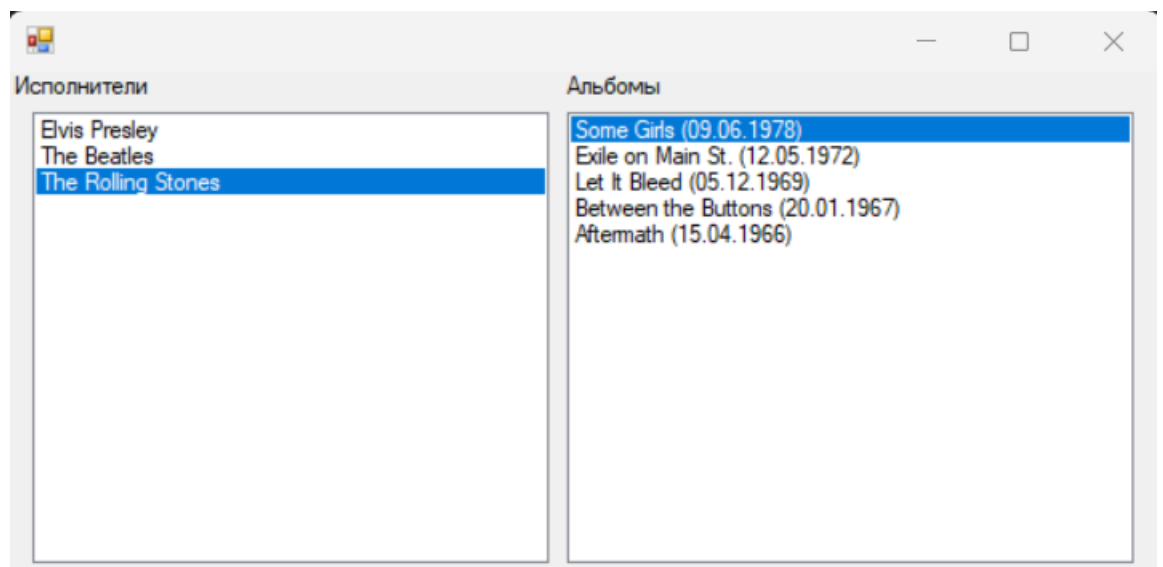


Рисунок 7 – Сортировка списка исполнителей и альбомов

5. Завершение работы и сохранение результатов

Код программы был проверен на наличие ошибок, и все ошибки были устранены. Графический интерфейс корректно отображал данные. Полученные листинги программы и выполненные изменения были сохранены для отчета.

ВЫВОД

По результатам выполнения работы было установлено, что использование LINQ позволяет эффективно обрабатывать коллекции данных. Реализация вывода уникального списка исполнителей с использованием методов `Select()` и `Distinct()` обеспечила автоматизацию обработки данных. Добавление функциональности фильтрации альбомов по выбранному исполнителю с помощью метода `Where()` и сортировки исполнителей и альбомов с использованием методов `OrderBy()` и `OrderByDescending()` позволило структурировать и упорядочить данные. Графический интерфейс корректно отображает исполнителей и их альбомы, обеспечивая удобство взаимодействия с программой.

ИСХОДНЫЙ КОД

Начальный код (Album.cs):

```
using System;
using System.Collections.Generic;

namespace OopLabs.Linq
{
    internal class Album
    {
        public string Artist { get; set; }
        public string Title { get; set; }
        public DateTime Date { get; set; }

        public Album(string artist, string title, DateTime date)
        {
            Artist = artist;
            Title = title;
            Date = date;
        }

        public override string ToString()
        {
            return string.Format("{0} ({1:dd.MM.yyyy})", Title, Date);
        }

        public static List<Album> GetAlbums()
        {
            return new List<Album>
            {
                new Album("The Beatles", "Revolver", new
DateTime(1966, 08, 05)),
                new Album("The Beatles", "Please Please Me", new
DateTime(1963, 03, 22)),
                new Album("The Beatles", "Abbey Road", new
DateTime(1969, 09, 26)),
                new Album("The Beatles", "Help!", new DateTime(1965,
08, 06)),
                new Album("The Beatles", "Rubber Soul", new
DateTime(1965, 12, 03)),
                new Album("The Beatles", "With The Beatles", new
DateTime(1963, 11, 22)),
                new Album("The Rolling Stones", "Some Girls", new
DateTime(1978, 06, 09)),
                new Album("The Rolling Stones", "Let It Bleed", new
DateTime(1969, 12, 05)),
                new Album("The Rolling Stones", "Exile on Main St.",
new DateTime(1972, 05, 12)),
                new Album("The Rolling Stones", "Aftermath", new
DateTime(1966, 04, 15)),
                new Album("The Rolling Stones", "Between the
Buttons", new DateTime(1967, 01, 20)),
                new Album("Elvis Presley", "How Great Thou Art", new
DateTime(1967, 02, 27)),
                new Album("Elvis Presley", "Elvis Is Back!", new
DateTime(1960, 04, 08)),
                new Album("Elvis Presley", "Elvis Country", new
DateTime(1971, 01, 02)),
                new Album("Elvis Presley", "Elvis Presley", new
DateTime(1956, 03, 23)),
            };
        }
    }
}
```



```
    }  
}
```

Упражнение 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;  
  
namespace OopLabs.Linq  
{  
    public partial class MainForm : Form  
    {  
  
        public MainForm()  
        {  
            InitializeComponent();  
        }  
  
        private void MainForm_Load(object sender, EventArgs e)  
        {  
            ArtistsListBox.DataSource = Album.GetAlbums()  
                .Select(album => album.Artist)  
                .Distinct()  
                .ToList();  
        }  
  
        private void InitializeComponent()  
        {  
            this.label1 = new System.Windows.Forms.Label();  
            this.ArtistsListBox = new System.Windows.Forms.ListBox();  
            this.SuspendLayout();  
            //  
            // label1  
            //  
            this.label1.AutoSize = true;  
            this.label1.Location = new System.Drawing.Point(0, 0);  
            this.label1.Name = "label1";  
            this.label1.Size = new System.Drawing.Size(95, 16);  
            this.label1.TabIndex = 0;  
            this.label1.Text = "Исполнители";  
            //  
            // ArtistsListBox  
            //  
            this.ArtistsListBox.BackColor = System.Drawing.SystemColors.Window;  
            this.ArtistsListBox.FormattingEnabled = true;  
            this.ArtistsListBox.ItemHeight = 16;  
            this.ArtistsListBox.Location = new System.Drawing.Point(8, 19);  
            this.ArtistsListBox.Name = "ArtistsListBox";  
            this.ArtistsListBox.Size = new System.Drawing.Size(270, 228);  
            this.ArtistsListBox.TabIndex = 1;  
            //  
            // MainForm  
            //  
            this.ClientSize = new System.Drawing.Size(282, 253);  
            this.Controls.Add(this.ArtistsListBox);  
            this.Controls.Add(this.label1);  
            this.Name = "MainForm";  
        }  
    }  
}
```

```

        this.Load += new System.EventHandler(this.MainForm_Load);
        this.ResumeLayout(false);
        this.PerformLayout();
    }
}

```

Упражнение 2:

```

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 OopLabs.Linq
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void MainForm_Load(object sender, EventArgs e)
            {
                ArtistsListBox.DataSource = Album.GetAlbums()
                    .Select(album => album.Artist)
                    .Distinct()
                    .ToList();
            }

            private void InitializeComponent()
            {
                this.label1 = new System.Windows.Forms.Label();
                this.ArtistsListBox = new System.Windows.Forms.ListBox();
                this.label2 = new System.Windows.Forms.Label();
                this.AlbumsListBox = new System.Windows.Forms.ListBox();
                this.SuspendLayout();
                //
                // label1
                //
                this.label1.AutoSize = true;
                this.label1.Location = new System.Drawing.Point(0, 0);
                this.label1.Name = "label1";
                this.label1.Size = new System.Drawing.Size(95, 16);
                this.label1.TabIndex = 0;
                this.label1.Text = "Исполнители";
                //
                // ArtistsListBox
                //
                this.ArtistsListBox.FormattingEnabled = true;
                this.ArtistsListBox.ItemHeight = 16;
                this.ArtistsListBox.Location = new System.Drawing.Point(12, 20);
                this.ArtistsListBox.Name = "ArtistsListBox";
                this.ArtistsListBox.Size = new System.Drawing.Size(258, 228);
                this.ArtistsListBox.TabIndex = 1;
            }
        }
    }
}

```

```

        this.ArtistsListBox.SelectedIndexChanged += new
System.EventHandler(this.ArtistsListBox_SelectedIndexChanged);
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Location = new System.Drawing.Point(276, 0);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(65, 16);
        this.label2.TabIndex = 2;
        this.label2.Text = "Альбомы";
        //
        // AlbumsListBox
        //
        this.AlbumsListBox.FormattingEnabled = true;
        this.AlbumsListBox.ItemHeight = 16;
        this.AlbumsListBox.Location = new System.Drawing.Point(279, 20);
        this.AlbumsListBox.Name = "AlbumsListBox";
        this.AlbumsListBox.Size = new System.Drawing.Size(283, 228);
        this.AlbumsListBox.TabIndex = 3;
        //
        // MainForm
        //
        this.ClientSize = new System.Drawing.Size(576, 253);
        this.Controls.Add(this.AlbumsListBox);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.ArtistsListBox);
        this.Controls.Add(this.label1);
        this.Name = "MainForm";
        this.Load += new System.EventHandler(this.MainForm_Load);
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    private void ArtistsListBox_SelectedIndexChanged(object sender,
EventArgs e)
    {
        AlbumsListBox.DataSource = Album.GetAlbums()
        .Where(album => album.Artist ==
ArtistsListBox.SelectedItem.ToString())
        .ToList();
    }
}

```

Упражнение 3:

```

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 OopLabs.Linq
{
    public partial class MainForm : Form
    {
        public MainForm()

```

```

{
    InitializeComponent();
}

private void MainForm_Load(object sender, EventArgs e)
{
    ArtistsListBox.DataSource = Album.GetAlbums()
        .Select(album => album.Artist)
        .OrderBy(artist => artist)
        .Distinct()
        .ToList();
}

private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.ArtistsListBox = new System.Windows.Forms.ListBox();
    this.label2 = new System.Windows.Forms.Label();
    this.AlbumsListBox = new System.Windows.Forms.ListBox();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.Location = new System.Drawing.Point(0, 0);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(95, 16);
    this.label1.TabIndex = 0;
    this.label1.Text = "Исполнители";
    //
    // ArtistsListBox
    //
    this.ArtistsListBox.FormattingEnabled = true;
    this.ArtistsListBox.ItemHeight = 16;
    this.ArtistsListBox.Location = new System.Drawing.Point(12, 20);
    this.ArtistsListBox.Name = "ArtistsListBox";
    this.ArtistsListBox.Size = new System.Drawing.Size(258, 228);
    this.ArtistsListBox.TabIndex = 1;
    this.ArtistsListBox.SelectedIndexChanged += new
System.EventHandler(this.ArtistsListBox_SelectedIndexChanged);
    //
    // label2
    //
    this.label2.AutoSize = true;
    this.label2.Location = new System.Drawing.Point(276, 0);
    this.label2.Name = "label2";
    this.label2.Size = new System.Drawing.Size(65, 16);
    this.label2.TabIndex = 2;
    this.label2.Text = "Альбомы";
    //
    // AlbumsListBox
    //
    this.AlbumsListBox.FormattingEnabled = true;
    this.AlbumsListBox.ItemHeight = 16;
    this.AlbumsListBox.Location = new System.Drawing.Point(279, 20);
    this.AlbumsListBox.Name = "AlbumsListBox";
    this.AlbumsListBox.Size = new System.Drawing.Size(283, 228);
    this.AlbumsListBox.TabIndex = 3;
    //
    // MainForm
    //
    this.ClientSize = new System.Drawing.Size(576, 253);
    this.Controls.Add(this.AlbumsListBox);
    this.Controls.Add(this.label2);
    this.Controls.Add(this.ArtistsListBox);
}

```

```

        this.Controls.Add(this.label1);
        this.Name = "MainForm";
        this.Load += new System.EventHandler(this.MainForm_Load);
        this.ResumeLayout(false);
        this.PerformLayout();
    }

    private void ArtistsListBox_SelectedIndexChanged(object sender,
EventArgs e)
    {
        AlbumsListBox.DataSource = Album.GetAlbums()
        .Where(album => album.Artist ==
ArtistsListBox.SelectedItem.ToString())
        .OrderByDescending(album =>
album.Date)
        .ToList();
    }
}

```