Лабораторная работа № 4.  
Работа с коллекциями данных  
с помощью LINQ

**Цель лабораторной работы**

Закрепление теоретических знаний по основам работы с коллекциями данных при помощи LINQ.

**Постановка задачи**

**Language Integrated Query (LINQ)** — это технология, интегрированная в язык C#, которая предоставляет унифицированный синтаксис для выполнения запросов к различным источникам данных: коллекциям в памяти, базам данных, XML-документам и другим. LINQ объединяет возможности декларативного программирования с императивным стилем C#, что повышает читаемость кода и снижает вероятность ошибок.

**Основные концепции LINQ**

**Определение и назначение**

LINQ позволяет выражать запросы непосредственно в коде на C#, используя синтаксис, напоминающий SQL. Это достигается за счет:

* **Расширяющих методов (extension methods)**, таких как Where(), Select(), OrderBy(), определенных в пространстве имен System.Linq.
* **Выражений запросов (query expressions)**, которые компилируются в вызовы этих методов.

**Источники данных LINQ**

LINQ поддерживает несколько провайдеров для работы с разными типами данных:

* **LINQ to Objects** — запросы к коллекциям в памяти (массивы, списки, словари).
* **LINQ to Entities / Entity Framework Core** — запросы к базам данных через ORM.
* **LINQ to XML** — работа с XML-документами.
* **LINQ to SQL** (устаревший) — ранняя реализация для SQL Server.
* 2. Синтаксис LINQ

**Методный синтаксис (Method Syntax)**

Основан на цепочках вызовов методов-расширений. Пример:

record Sample(int Id,

string Name,

int Age,

int Property,

string Department);

var collection = new List<Sample>();

var result = collection

.Where(item => item.Property > 10)

.OrderBy(item => item.Name)

.Select(item => new { item.Id, item.Name });

**Синтаксис запросов (Query Syntax)**

Напоминает SQL и компилируется в методные вызовы:

var result = from item in collection

where item.Property > 10

orderby item.Name

select new { item.Id, item.Name };

Различия и выбор синтаксиса

* **Query Syntax** удобен для простых запросов, особенно с join и group by.
* **Method Syntax** более гибкий, поддерживает все операции и часто используется в комбинации с лямбда-выражениями.

Основные операции LINQ

Фильтрация

* Where() — выбирает элементы, удовлетворяющие условию.

var filtered = collection.Where(x => x.Age > 18);

Проекция

* Select() — преобразует элементы в новую форму.

var names = collection.Select(x => x.Name);

Сортировка

* OrderBy(), OrderByDescending(), ThenBy() — сортировка по ключу.

var sorted = collection.OrderBy(x => x.Age).ThenBy(x => x.Name);

Группировка

* GroupBy() — группирует элементы по ключу.

var groups = collection.GroupBy(x => x.Department);

Агрегация

* Count(), Sum(), Average(), Max(), Min() — статистические операции.

var total = collection.Sum(x => x.Price);

Соединения (Joins)

* Join() и GroupJoin() — аналоги SQL INNER JOIN и LEFT JOIN.

var joined = customers.Join(orders,

c => c.Id,

o => o.CustomerId,

(c, o) => new { c.Name, o.Amount });

**Варианты заданий на лабораторную работу**

1. Необходимо разработать информационную систему с графическим интерфейсом в WindowsForms.
2. Программа должна быть загружена в GitHub репозиторий.
3. Предусмотреть формы для CRUD-операций (Create, Read, Update, Delete) с заданными классами.
4. Операции должны обязательно использовать LINQ. Предусмотреть: фильтрация данных, сортировка по определенному критерию, проекция, группировка и агрегация.
5. Предусмотреть валидацию ввода (проверка на пустые поля и др.).
6. **Для оценки «Удовлетворительно».** Использовать 1 класс.
7. **Для оценки «Хорошо».** Использовать 2 класса.
8. **Для оценки «Отлично».** Использовать 3 класса.
9. Защита работы включает демонстрацию работы программы.
10. Для успешной защиты:

* программа должна быть задокументирована с помощью комментариев,
* код программы не должен содержать не используемые блоки, должен быть «чистым» и стилистически верно оформлен,
* интерфейс программы должен быть гибким и удобным.

**Варианты заданий на лабораторную работу**

1. Музыкальная коллекция

* Классы: Song, Artist, Album.

1. Фильмотека

* Классы: Movie, Director, Studio.

1. Журналы и статьи

* Классы: Article, Journal, Category.

1. Аудиокниги

* Классы: Audiobook, Narrator, Publisher.

1. Комиксы

* Классы: Comic, Writer, Series.

1. Видеоигры

* Классы: Game, Developer, Platform.

1. Научные работы

* Классы: ResearchPaper, Scientist, Field.

1. Фотогалерея

* Классы: Photo, Photographer, Location.

1. Коллекция рецептов

* Классы: Recipe, Chef, Cuisine.

1. Учёт студентов

* Классы: Student, Group, Faculty.

1. Расписание занятий

* Классы: Lesson, Teacher, Classroom.

1. Спортивные секции

* Классы: Athlete, Coach, Sport.

1. Школьные предметы

* Классы: Subject, Teacher, Grade.

1. Курсы онлайн-обучения

* Классы: Course, Instructor, Topic.

1. Турниры по играм

* Классы: Tournament, Player, Game.

1. Фитнес-клуб

* Классы: Client, Trainer, Membership.

1. Музыкальная школа

* Классы: Student, Instrument, Teacher.

1. Библиотека учебников

* Классы: Textbook, Subject, Publisher.

1. Шахматный клуб

* Классы: Player, Tournament, Club.

1. Учёт сотрудников

* Классы: Employee, Department, Position.

1. Заказы в ресторане

* Классы: Order, Dish, Waiter.

1. Автосервис

* Классы: Car, Owner, Service.

1. Аптека

* Классы: Medicine, Manufacturer, Category.

1. Гостиница

* Классы: Guest, Room, Booking.

1. Таксопарк

* Классы: Driver, Car, Order.

1. Агентство недвижимости

* Классы: Property, Owner, Agent.

1. Фитнес-трекер

* Классы: Workout, User, Exercise.

1. Салон красоты

* Классы: Client, Service, Master.

1. Фермерское хозяйство

* Классы: Animal, Breed, Farm.

**Пример разработки приложения**

**1. Описание предметной области**

Будем использовать три класса:

1. Book (Книга) – содержит информацию о книге.
2. Author (Автор) – хранит данные об авторе.
3. Genre (Жанр) – определяет жанр книги.

**2. Создание проекта и настройка**

1. Откройте Visual Studio → **Создать проект** → **Windows Forms App (.NET Framework)**.
2. Назовите проект **"LibrarySystem"**.
3. Убедитесь, что выбран **.NET 8.0**.

Добавляем папки: Domain, Repositories, Forms, Services.

Архитектура приложения

LibrarySystem/

├── Domain/ # Бизнес-логика

│ ├── Author.cs

│ ├── Genre.cs

│ └── Book.cs

├── Repositories/ # Работа с данными

│ └── BookRepository.cs

├── Forms/ # Пользовательский интерфейс

│ ├── MainForm.cs

│ └── BookEditForm.cs

└── Services/ # Доменные сервисы

└── BookService.cs

**3. Реализация моделей**

Domain/Genre.cs

namespace LibrarySystem.Domain;

public class Genre

{

public int Id { get; }

public string Name { get; }

public Genre(int id, string name)

{

if (string.IsNullOrWhiteSpace(name))

throw new ArgumentException("Название жанра не может быть пустым");

Id = id;

Name = name;

}

public override string ToString() => Name;

}

Domain/Author.cs

namespace LibrarySystem.Domain;

public class Author

{

public int Id { get; }

public string Name { get; }

public string Country { get; }

public Author(int id, string name, string country)

{

if (string.IsNullOrWhiteSpace(name))

throw new ArgumentException("Имя автора не может быть пустым");

Id = id;

Name = name;

Country = country;

}

public override string ToString() => $"{Name} ({Country})";

}

Domain/Book.cs

namespace LibrarySystem.Domain;

public class Book

{

public int Id { get; }

public string Title { get; private set; }

public Author Author { get; private set; }

public Genre Genre { get; private set; }

public int Year { get; private set; }

public decimal Price { get; private set; }

public Book(int id, string title, Author author, Genre genre, int year, decimal price)

{

Validate(title, year, price);

Id = id;

Title = title;

Author = author;

Genre = genre;

Year = year;

Price = price;

}

public void Update(string title, Author author, Genre genre, int year, decimal price)

{

Validate(title, year, price);

Title = title;

Author = author;

Genre = genre;

Year = year;

Price = price;

}

private void Validate(string title, int year, decimal price)

{

if (string.IsNullOrWhiteSpace(title))

throw new ArgumentException("Название книги не может быть пустым");

if (year < 1800 || year > DateTime.Now.Year + 1)

throw new ArgumentException("Некорректный год издания");

if (price <= 0)

throw new ArgumentException("Цена должна быть положительной");

}

public override string ToString() => $"{Title} ({Year})";

}

**4. Реализация BookRepository**

Repositories/BookRepository.cs

using LibrarySystem.Domain;

namespace LibrarySystem.Repositories;

public class BookRepository

{

private readonly List<Book> \_books = new();

private static int \_nextId = 1;

public void Add(string title, Author author, Genre genre, int year, decimal price)

{

\_books.Add(new Book(\_nextId, title, author, genre, year, price));

}

public void Update(Book book)

{

var existing = \_books.FirstOrDefault(b => b.Id == book.Id);

if (existing != null)

{

\_books.Remove(existing);

\_books.Add(book);

}

}

public void Delete(int id) => \_books.RemoveAll(b => b.Id == id);

public IEnumerable<Book> GetAll() => \_books.AsReadOnly();

// LINQ-методы

public IEnumerable<Book> GetByAuthor(int authorId) =>

\_books.Where(b => b.Author.Id == authorId);

public IEnumerable<Book> GetExpensiveBooks(decimal minPrice) =>

\_books.Where(b => b.Price >= minPrice);

public IEnumerable<IGrouping<Genre, Book>> GroupByGenre() =>

\_books.GroupBy(b => b.Genre);

}

**5. Реализация BookService**

Services/BookService.cs

using LibrarySystem.Domain;

using LibrarySystem.Repositories;

namespace LibrarySystem.Services;

public class BookService

{

private readonly BookRepository \_repository;

public BookService(BookRepository repository)

{

\_repository = repository;

}

public void AddBook(string title, Author author, Genre genre, int year, decimal price)

{

\_repository.Add(title, author, genre, year, price);

}

public IEnumerable<Book> GetAllBooks() => \_repository.GetAll();

public IEnumerable<Book> SearchBooks(string query) =>

\_repository.GetAll()

.Where(b => b.Title.Contains(query, StringComparison.OrdinalIgnoreCase))

.OrderBy(b => b.Title);

public IEnumerable<Book> GetBooksSortedByYear(bool ascending = true) =>

ascending

? \_repository.GetAll().OrderBy(b => b.Year)

: \_repository.GetAll().OrderByDescending(b => b.Year);

public IEnumerable<IGrouping<Genre, Book>> GetBooksGroupedByGenre() =>

\_repository.GroupByGenre();

}

**6. Реализация главной формы**

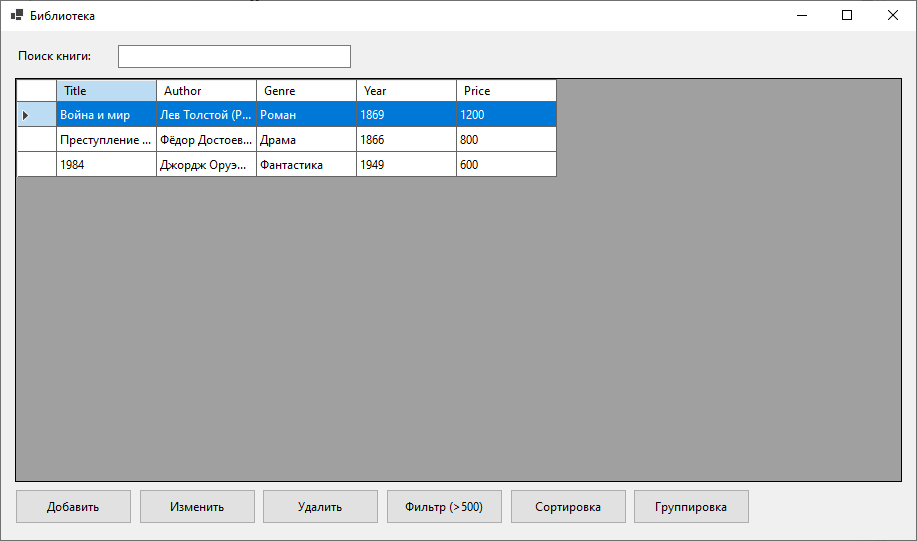


Рисунок 1 – Главная форма

Forms/MainForm.cs

using LibrarySystem.Domain;

using LibrarySystem.Repositories;

using LibrarySystem.Services;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace LibrarySystem.Forms

{

public partial class MainForm : Form

{

private readonly BookService \_bookService;

private readonly List<Author> \_authors;

private readonly List<Genre> \_genres;

public MainForm()

{

InitializeComponent();

\_bookService = new BookService(new BookRepository());

\_authors = LoadSampleAuthors();

\_genres = LoadSampleGenres();

LoadSampleBooks();

RefreshDataGrid();

}

private List<Author> LoadSampleAuthors() => new()

{

new Author(1, "Лев Толстой", "Россия"),

new Author(2, "Фёдор Достоевский", "Россия"),

new Author(3, "Джордж Оруэлл", "Великобритания")

};

private List<Genre> LoadSampleGenres() => new()

{

new Genre(1, "Роман"),

new Genre(2, "Фантастика"),

new Genre(3, "Драма")

};

private void LoadSampleBooks()

{

\_bookService.AddBook("Война и мир", \_authors[0], \_genres[0], 1869, 1200);

\_bookService.AddBook("Преступление и наказание", \_authors[1], \_genres[2], 1866, 800);

\_bookService.AddBook("1984", \_authors[2], \_genres[1], 1949, 600);

}

private void RefreshDataGrid(IEnumerable<Book> books = null)

{

dataGridView.DataSource = (books ?? \_bookService.GetAllBooks()).ToList();

dataGridView.Columns["Id"].Visible = false;

}

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

{

using var form = new BookEditForm(null, \_authors, \_genres);

if (form.ShowDialog() == DialogResult.OK)

{

try

{

\_bookService.AddBook(

form.BookTitle,

form.SelectedAuthor,

form.SelectedGenre,

form.Year,

form.Price);

RefreshDataGrid();

}

catch (Exception ex)

{

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

}

}

}

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

{

if (dataGridView.SelectedRows.Count == 0) return;

var selectedBook = (Book)dataGridView.SelectedRows[0].DataBoundItem;

using var form = new BookEditForm(selectedBook, \_authors, \_genres);

if (form.ShowDialog() == DialogResult.OK)

{

try

{

selectedBook.Update(

form.BookTitle,

form.SelectedAuthor,

form.SelectedGenre,

form.Year,

form.Price);

\_bookService.UpdateBook(selectedBook);

RefreshDataGrid();

}

catch (Exception ex)

{

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

}

}

}

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

{

if (dataGridView.SelectedRows.Count == 0) return;

var selectedBook = (Book)dataGridView.SelectedRows[0].DataBoundItem;

if (MessageBox.Show(

$"Удалить книгу '{selectedBook.Title}'?",

"Подтверждение",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question) == DialogResult.Yes)

{

\_bookService.DeleteBook(selectedBook.Id);

RefreshDataGrid();

}

}

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

{

RefreshDataGrid(\_bookService.GetExpensiveBooks(500));

}

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

{

RefreshDataGrid(\_bookService.GetBooksSortedByYear());

}

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

{

var groupedBooks = \_bookService.GetBooksGroupedByGenre()

.SelectMany(g => g.Select(b => new

{

Жанр = g.Key.Name,

Книга = b.Title,

Автор = b.Author.Name,

Год = b.Year,

Цена = b.Price

}))

.ToList();

dataGridView.DataSource = groupedBooks;

}

private void txtSearch\_TextChanged(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtSearch.Text))

RefreshDataGrid();

else

RefreshDataGrid(\_bookService.SearchBooks(txtSearch.Text));

}

}

}

Forms/MainForm.Designer.cs

namespace LibrarySystem.Forms

{

partial class MainForm

{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

private void InitializeComponent()

{

this.dataGridView = new System.Windows.Forms.DataGridView();

this.btnAdd = new System.Windows.Forms.Button();

this.btnEdit = new System.Windows.Forms.Button();

this.btnDelete = new System.Windows.Forms.Button();

this.btnFilter = new System.Windows.Forms.Button();

this.btnSort = new System.Windows.Forms.Button();

this.btnGroup = new System.Windows.Forms.Button();

this.txtSearch = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

((System.ComponentModel.ISupportInitialize)(this.dataGridView)).BeginInit();

this.SuspendLayout();

//

// dataGridView

//

this.dataGridView.AllowUserToAddRows = false;

this.dataGridView.AllowUserToDeleteRows = false;

this.dataGridView.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

this.dataGridView.Location = new System.Drawing.Point(12, 41);

this.dataGridView.MultiSelect = false;

this.dataGridView.Name = "dataGridView";

this.dataGridView.ReadOnly = true;

this.dataGridView.SelectionMode = System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;

this.dataGridView.Size = new System.Drawing.Size(760, 350);

this.dataGridView.TabIndex = 0;

//

// btnAdd

//

this.btnAdd.Location = new System.Drawing.Point(12, 397);

this.btnAdd.Name = "btnAdd";

this.btnAdd.Size = new System.Drawing.Size(100, 30);

this.btnAdd.TabIndex = 1;

this.btnAdd.Text = "Добавить";

this.btnAdd.UseVisualStyleBackColor = true;

this.btnAdd.Click += new System.EventHandler(this.btnAdd\_Click);

//

// btnEdit

//

this.btnEdit.Location = new System.Drawing.Point(118, 397);

this.btnEdit.Name = "btnEdit";

this.btnEdit.Size = new System.Drawing.Size(100, 30);

this.btnEdit.TabIndex = 2;

this.btnEdit.Text = "Изменить";

this.btnEdit.UseVisualStyleBackColor = true;

this.btnEdit.Click += new System.EventHandler(this.btnEdit\_Click);

//

// btnDelete

//

this.btnDelete.Location = new System.Drawing.Point(224, 397);

this.btnDelete.Name = "btnDelete";

this.btnDelete.Size = new System.Drawing.Size(100, 30);

this.btnDelete.TabIndex = 3;

this.btnDelete.Text = "Удалить";

this.btnDelete.UseVisualStyleBackColor = true;

this.btnDelete.Click += new System.EventHandler(this.btnDelete\_Click);

//

// btnFilter

//

this.btnFilter.Location = new System.Drawing.Point(330, 397);

this.btnFilter.Name = "btnFilter";

this.btnFilter.Size = new System.Drawing.Size(100, 30);

this.btnFilter.TabIndex = 4;

this.btnFilter.Text = "Фильтр (>500)";

this.btnFilter.UseVisualStyleBackColor = true;

this.btnFilter.Click += new System.EventHandler(this.btnFilter\_Click);

//

// btnSort

//

this.btnSort.Location = new System.Drawing.Point(436, 397);

this.btnSort.Name = "btnSort";

this.btnSort.Size = new System.Drawing.Size(100, 30);

this.btnSort.TabIndex = 5;

this.btnSort.Text = "Сортировка";

this.btnSort.UseVisualStyleBackColor = true;

this.btnSort.Click += new System.EventHandler(this.btnSort\_Click);

//

// btnGroup

//

this.btnGroup.Location = new System.Drawing.Point(542, 397);

this.btnGroup.Name = "btnGroup";

this.btnGroup.Size = new System.Drawing.Size(100, 30);

this.btnGroup.TabIndex = 6;

this.btnGroup.Text = "Группировка";

this.btnGroup.UseVisualStyleBackColor = true;

this.btnGroup.Click += new System.EventHandler(this.btnGroup\_Click);

//

// txtSearch

//

this.txtSearch.Location = new System.Drawing.Point(100, 12);

this.txtSearch.Name = "txtSearch";

this.txtSearch.Size = new System.Drawing.Size(200, 20);

this.txtSearch.TabIndex = 7;

this.txtSearch.TextChanged += new System.EventHandler(this.txtSearch\_TextChanged);

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(12, 15);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(82, 13);

this.label1.TabIndex = 8;

this.label1.Text = "Поиск книги:";

//

// MainForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(784, 441);

this.Controls.Add(this.label1);

this.Controls.Add(this.txtSearch);

this.Controls.Add(this.btnGroup);

this.Controls.Add(this.btnSort);

this.Controls.Add(this.btnFilter);

this.Controls.Add(this.btnDelete);

this.Controls.Add(this.btnEdit);

this.Controls.Add(this.btnAdd);

this.Controls.Add(this.dataGridView);

this.Name = "MainForm";

this.Text = "Библиотека";

((System.ComponentModel.ISupportInitialize)(this.dataGridView)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.DataGridView dataGridView;

private System.Windows.Forms.Button btnAdd;

private System.Windows.Forms.Button btnEdit;

private System.Windows.Forms.Button btnDelete;

private System.Windows.Forms.Button btnFilter;

private System.Windows.Forms.Button btnSort;

private System.Windows.Forms.Button btnGroup;

private System.Windows.Forms.TextBox txtSearch;

private System.Windows.Forms.Label label1;

}

}

**7. Реализация формы редактирования**

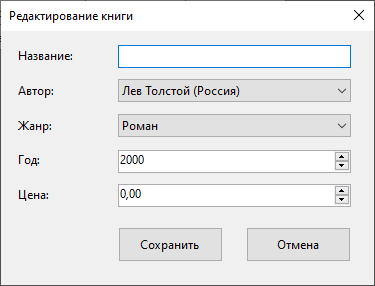


Рисунок 2 – Форма редактирования

Forms/BookEditForm.cs

using LibrarySystem.Domain;

using System;

using System.Collections.Generic;

using System.Windows.Forms;

using static System.Runtime.InteropServices.JavaScript.JSType;

namespace LibrarySystem.Forms

{

public partial class BookEditForm : Form

{

public string BookTitle => txtTitle.Text;

public Author SelectedAuthor => (Author)cmbAuthor.SelectedItem;

public Genre SelectedGenre => (Genre)cmbGenre.SelectedItem;

public int Year => (int)numYear.Value;

public decimal Price => numPrice.Value;

public BookEditForm(Book book, List<Author> authors, List<Genre> genres)

{

InitializeComponent();

cmbAuthor.DataSource = authors;

cmbGenre.DataSource = genres;

if (book != null)

{

txtTitle.Text = book.Title;

cmbAuthor.SelectedItem = authors.Find(a => a.Id == book.Author.Id);

cmbGenre.SelectedItem = genres.Find(g => g.Id == book.Genre.Id);

numYear.Value = book.Year;

numPrice.Value = book.Price;

}

}

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

{

if (!ValidateForm())

return;

DialogResult = DialogResult.OK;

Close();

}

private bool ValidateForm()

{

if (string.IsNullOrWhiteSpace(txtTitle.Text))

{

MessageBox.Show("Название книги не может быть пустым", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

if (numYear.Value < 1800 || numYear.Value > DateTime.Now.Year + 1)

{

MessageBox.Show($"Год должен быть между 1800 и {DateTime.Now.Year + 1}",

"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

if (numPrice.Value <= 0)

{

MessageBox.Show("Цена должна быть положительной", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

return true;

}

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

{

DialogResult = DialogResult.Cancel;

Close();

}

}

}

Forms/BookEditForm.Designer.cs

namespace LibrarySystem.Forms

{

partial class BookEditForm

{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

private void InitializeComponent()

{

this.label1 = new System.Windows.Forms.Label();

this.txtTitle = new System.Windows.Forms.TextBox();

this.label2 = new System.Windows.Forms.Label();

this.cmbAuthor = new System.Windows.Forms.ComboBox();

this.label3 = new System.Windows.Forms.Label();

this.cmbGenre = new System.Windows.Forms.ComboBox();

this.label4 = new System.Windows.Forms.Label();

this.numYear = new System.Windows.Forms.NumericUpDown();

this.label5 = new System.Windows.Forms.Label();

this.numPrice = new System.Windows.Forms.NumericUpDown();

this.btnSave = new System.Windows.Forms.Button();

this.btnCancel = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.numYear)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.numPrice)).BeginInit();

this.SuspendLayout();

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(12, 15);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(60, 13);

this.label1.TabIndex = 0;

this.label1.Text = "Название:";

//

// txtTitle

//

this.txtTitle.Location = new System.Drawing.Point(100, 12);

this.txtTitle.Name = "txtTitle";

this.txtTitle.Size = new System.Drawing.Size(200, 20);

this.txtTitle.TabIndex = 1;

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(12, 45);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(40, 13);

this.label2.TabIndex = 2;

this.label2.Text = "Автор:";

//

// cmbAuthor

//

this.cmbAuthor.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

this.cmbAuthor.FormattingEnabled = true;

this.cmbAuthor.Location = new System.Drawing.Point(100, 42);

this.cmbAuthor.Name = "cmbAuthor";

this.cmbAuthor.Size = new System.Drawing.Size(200, 21);

this.cmbAuthor.TabIndex = 3;

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(12, 75);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(39, 13);

this.label3.TabIndex = 4;

this.label3.Text = "Жанр:";

//

// cmbGenre

//

this.cmbGenre.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

this.cmbGenre.FormattingEnabled = true;

this.cmbGenre.Location = new System.Drawing.Point(100, 72);

this.cmbGenre.Name = "cmbGenre";

this.cmbGenre.Size = new System.Drawing.Size(200, 21);

this.cmbGenre.TabIndex = 5;

//

// label4

//

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(12, 105);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(28, 13);

this.label4.TabIndex = 6;

this.label4.Text = "Год:";

//

// numYear

//

this.numYear.Location = new System.Drawing.Point(100, 103);

this.numYear.Maximum = new decimal(new int[] {

2100,

0,

0,

0});

this.numYear.Minimum = new decimal(new int[] {

1800,

0,

0,

0});

this.numYear.Name = "numYear";

this.numYear.Size = new System.Drawing.Size(200, 20);

this.numYear.TabIndex = 7;

this.numYear.Value = new decimal(new int[] {

2000,

0,

0,

0});

//

// label5

//

this.label5.AutoSize = true;

this.label5.Location = new System.Drawing.Point(12, 135);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(36, 13);

this.label5.TabIndex = 8;

this.label5.Text = "Цена:";

//

// numPrice

//

this.numPrice.DecimalPlaces = 2;

this.numPrice.Location = new System.Drawing.Point(100, 133);

this.numPrice.Maximum = new decimal(new int[] {

100000,

0,

0,

0});

this.numPrice.Name = "numPrice";

this.numPrice.Size = new System.Drawing.Size(200, 20);

this.numPrice.TabIndex = 9;

//

// btnSave

//

this.btnSave.Location = new System.Drawing.Point(100, 170);

this.btnSave.Name = "btnSave";

this.btnSave.Size = new System.Drawing.Size(90, 30);

this.btnSave.TabIndex = 10;

this.btnSave.Text = "Сохранить";

this.btnSave.UseVisualStyleBackColor = true;

this.btnSave.Click += new System.EventHandler(this.btnSave\_Click);

//

// btnCancel

//

this.btnCancel.Location = new System.Drawing.Point(210, 170);

this.btnCancel.Name = "btnCancel";

this.btnCancel.Size = new System.Drawing.Size(90, 30);

this.btnCancel.TabIndex = 11;

this.btnCancel.Text = "Отмена";

this.btnCancel.UseVisualStyleBackColor = true;

this.btnCancel.Click += new System.EventHandler(this.btnCancel\_Click);

//

// BookEditForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(320, 220);

this.Controls.Add(this.btnCancel);

this.Controls.Add(this.btnSave);

this.Controls.Add(this.numPrice);

this.Controls.Add(this.label5);

this.Controls.Add(this.numYear);

this.Controls.Add(this.label4);

this.Controls.Add(this.cmbGenre);

this.Controls.Add(this.label3);

this.Controls.Add(this.cmbAuthor);

this.Controls.Add(this.label2);

this.Controls.Add(this.txtTitle);

this.Controls.Add(this.label1);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;

this.MaximizeBox = false;

this.MinimizeBox = false;

this.Name = "BookEditForm";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;

this.Text = "Редактирование книги";

((System.ComponentModel.ISupportInitialize)(this.numYear)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numPrice)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Label label1;

private System.Windows.Forms.TextBox txtTitle;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.ComboBox cmbAuthor;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.ComboBox cmbGenre;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.NumericUpDown numYear;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.NumericUpDown numPrice;

private System.Windows.Forms.Button btnSave;

private System.Windows.Forms.Button btnCancel;

}

}

# Содержание пояснительной записки

1. Постановка задачи. Приводится теоретический материал, использованный при написании приложения.

2. Формулировка задания и вариант. Приводится задание на лабораторную работу и вариант этого задания.

3. Описание выполняемых действий. Необходимо привести описание последовательности разработки программы, реализации используемых методов, алгоритмов, блок-схем.

4. Анализ результатов. Привести анализ входных и выходных данных. Показать результаты выполнения программного кода. Предоставить скриншоты обработки тестовых примеров. Сделать выводы.

5. Листинг программы. Привести листинг разработанного программного кода, содержание файлов входных и выходных данных.

# Используемое программное обеспечение

1. Среда программирования MS Visual Studio Community 2022 (Свободно распространяемое программное обеспечение (в учебных целях));
2. Microsoft Office Standard 2007 (Open License: 42267924);
3. Open Office (Свободно распространяемое программное обеспечение).
4. Браузер (Свободно распространяемое программное обеспечение).

# Список литературы

* + - 1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 285 c.
      2. Биллиг, В. A. Основы объектного программирования на С# (C# 3.0, Visual Studio 2008) [Электронный ресурс] : учебное пособие / В. A. Биллиг. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 583 c. — 978-5-4487-0145-0. — Режим доступа: http://www.iprbookshop.ru/72339.html
      3. Павловская, Т. А. Программирование на языке высокого уровня C# [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html
      4. Агапов, В. П. Основы программирования на языке С# [Электронный ресурс] : учебное пособие / В. П. Агапов. — Электрон. текстовые данные. — М. : Московский государственный строительный университет, ЭБС АСВ, 2012. — 128 c. — 978-5-7264-0576-6. — Режим доступа: http://www.iprbookshop.ru/16366.html
      5. Медведев, М. А. Программирование на СИ# [Электронный ресурс] : учебное пособие / М. А. Медведев, А. Н. Медведев ; под ред. А. В. Присяжный. — Электрон. текстовые данные. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2015. — 64 c. — 978-5-7996-1561-1. — Режим доступа: http://www.iprbookshop.ru/69667.html
      6. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual С# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2011.— 180 c
      7. Уйманова Н.А. Основы объектно-ориентированного программирования [Электронный ресурс]: практикум/ Уйманова Н.А., Таспаева М.Г.— Электрон. текстовые данные.— Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2017.— 156 c.— Режим доступа: http://www.iprbookshop.ru/78808.html.— ЭБС «IPRbooks»
      8. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ Новиков П.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 c.— Режим доступа: http://www.iprbookshop.ru/64650.html.— ЭБС «IPRbooks»