**ФГБОУ ВО**

**Национальный исследовательский университет**

**«МЭИ»**

**Лабораторная работа № 6**

**Множественное наследование**

**Вариант 6.**

**Выполнил:**

Студент группы Аэ-21-23

Ицков И. С.

**Москва, 2024**

*Условие:*

1. Построить иерархию классов в соответствии с вариантом задания.
2. Разработать методы и свойства для каждого из определяемых классов.
3. Реализовать программу на C# в соответствии с вариантом задания c использованием Windows From.

**1. Постановка задачи.**

*Задание:*

Создать иерархию классов. Описать их поля и методы. Иерархия классов: Журнал, книга, печатное издание, учебник. В программе должна выполняться проверка всех разработанных элементов классов.

Требуется создать программу, которая обладает следующими функциями:

* Возможность взаимодействия с классом Журнал
* Возможность взаимодействия с классом Книга
* Возможность взаимодействия с классом Учебник
* Выход

В качестве входных данных программа принимает на вход: • Целые числа:

* Количество страниц публикации
* Номер выпуска (для журналов и учебников) • Строки:
* Название публикации
* Имя автора (для книг и учебников)
* Предмет (для учебников)

В качестве выходных данных программа выводит результат взаимодействия с объектом одного из трёх классов: • Журнал (Magazine) • Книга (Book) • Учебник (Textbook)

Необходимо предусмотреть: • Установлены границы для количества страниц (не может быть отрицательным или превышать 10000 страниц). При превышении границы выводится ошибка, пользователь должен исправить вводимые данные.

• Номер выпуска журнала или учебника должен быть положительным числом и не может превышать 1000. При вводе некорректного номера выводится ошибка, пользователь должен исправить вводимые данные.

• Возможно выбрать только три типа публикации через выпадающий список: "Журнал", "Книга", "Учебник". Это предотвращает ввод некорректного типа публикации пользователем.

• Название публикации и имя автора не могут быть пустыми строками или содержать только пробелы. При попытке ввода некорректных данных выводится ошибка, пользователь должен исправить вводимые данные.

• Для учебников обязательно указание предмета, который не может быть пустым. В случае отсутствия предмета выводится ошибка, пользователь должен исправить вводимые данные.

**2. Разработка приложения**

2.1. Разработка структуры приложения

В программе написаны 5 классов, 3 из которых используют наследование, так же написаны интерфейсы, для реализации наследования . Классы содержат поля и методы. В итоге диаграмма классов имеет вид:



Рис 1. – Диаграмма классов

Описание полей и методов:

Для всех классов созданы интерфейсы, в которых в виде чисто виртуальных функций описаны методы, которые будут использоваться в классах, позже они переопределяются.

*Класс AbstractPublication:*

Это абстрактный базовый класс, от которого наследуются остальные классы публикаций.

Табл. 1. Описание класса AbstractPublication

|  |  |
| --- | --- |
| **Поля/Методы/Конструкторы** | **Описание** |
| Display() | Абстрактный метод для отображения ин формации о публикации. Переопределяется в производных классах |
| GetPages() | Абстрактный метод для получения количества страниц публикации. Возвращает значение целого типа |
| ComparePages(AbstractPublication pub1, AbstractPublication pub2) | Статический метод для сравнения количества страниц двух публикаций. Принимает две публикации и выводит результат сравнения |

*PrintPublication:*

Базовый класс для всех печатных изданий, наследуется от AbstractPublication.

Табл. 2. Описание класса AbstractPublication

|  |  |
| --- | --- |
| **Поля/Методы/Конструкторы** | **Описание** |
| PrintPublication(string title, int pages) | Параметризированный конструктор. Принимает название и количество страниц |
| Display() | Переопределенный метод для отображения базовой информации о публикации |
| GetPages() | Переопределенный метод для получения количества страниц |

*Класс MagazineBase:*

Класс для работы с журналами, наследуется от PrintPublication и реализует интерфейс IMagazine.

Табл. 3. Описание класса *MagazineBase*

|  |  |
| --- | --- |
| **Поля/Методы/Конструкторы** | **Описание** |
| MagazineBase(string title, int pages, int issueNumber) | Параметризированный конструктор |
| Driver(string n, int w, string type, int rl, int s, int cw) : Human(n, w) | Параметризированный конструктор |
| Display() | Переопределенный метод для отображения информации о журнале |

*Класс BookBase:*

Класс для работы с книгами, наследуется от PrintPublication и реализует интерфейс IBook.

Табл. 4. Описание класса BookBase

|  |  |
| --- | --- |
| **Поля/Методы/Конструкторы** | **Описание** |
| BookBase(string title, int pages, string author) | Параметризированный конструктор |
| Display() | Переопределенный метод для отображения информации о книге |
| DisplayBookInfo() | Метод для отображения специфической информации книги |

*Класс Textbook****:***

Класс для работы с учебниками, наследуется от PrintPublication и реализует оба интерфейса IMagazine и IBook.

Табл. 5. Описание класса Textbook

|  |  |
| --- | --- |
| **Поля/Методы/Конструкторы** | **Описание** |
| Textbook(string title, int pages, string author, int issueNumber, string subject) | Параметризированный конструктор |
| Display() | Переопределенный метод для отображения информации о книге |
| DisplayMagazineInfo() | Реализация метода интерфейса IMagazine |
| DisplayBookInfo() | Реализация метода интерфейса IBook |

2.2. Разработка схемы алгоритма

-

2.3. Разработка пользовательского интерфейса

Категория пользователей: Преподаватели и студенты

Функции пользователя: Приложение создано с использованием Windows Form, т.е. создано оконное приложение. Для его создания были использованы различные поля для ввода, кнопки и тд. В таблице ниже приведено подробное описание какие элементы управления были использованы и для чего. Пользователь может взаимодействовать с каждым элементом управления, для переключения между объектами классов или для ввода информации об этих же объектах.

Табл. 6. Описание элементов управления

|  |  |  |
| --- | --- | --- |
| Элемент Управления | Класс | Описание |
| publicationTypeCombo | ComboBox | Выпадающий список для выбора типа публикации ("Журнал", "Книга", "Учебник") |
| titleTextBox | TextBox | Окно для ввода названия публикации |
| pagesTextBox | TextBox | Окно для ввода количества страниц |
| authorTextBox | TextBox | Окно для ввода имени автора (для книг и учебников) |
| issueNumberTextBox | TextBox | Окно для ввода номера выпуска (для журналов и учебников) |
| subjectTextBox | TextBox | Окно для ввода предмета (для учебников) |
| titleLabel | Label | Надпись "Название:" |
| pagesLabel | Label | Надпись "Страницы:" |
| authorLabel | Label | Надпись "Автор:" |
| issueNumberLabel | Label | Надпись "Номер выпуска:" |
| subjectLabel | Label | Надпись "Предмет:" |
| addButton | Button | Кнопка для добавления новой публикации в список |
| displayAllButton | Button | Кнопка для отображения всех добавленных публикаций |
| displayTextBox | RichTextBox | Окно для вывода информации о публикациях |
| inputPanel | TableLayoutPanel | Панель для организации элементов ввода в табличном формате |

Примеры диалогов с пользователем:

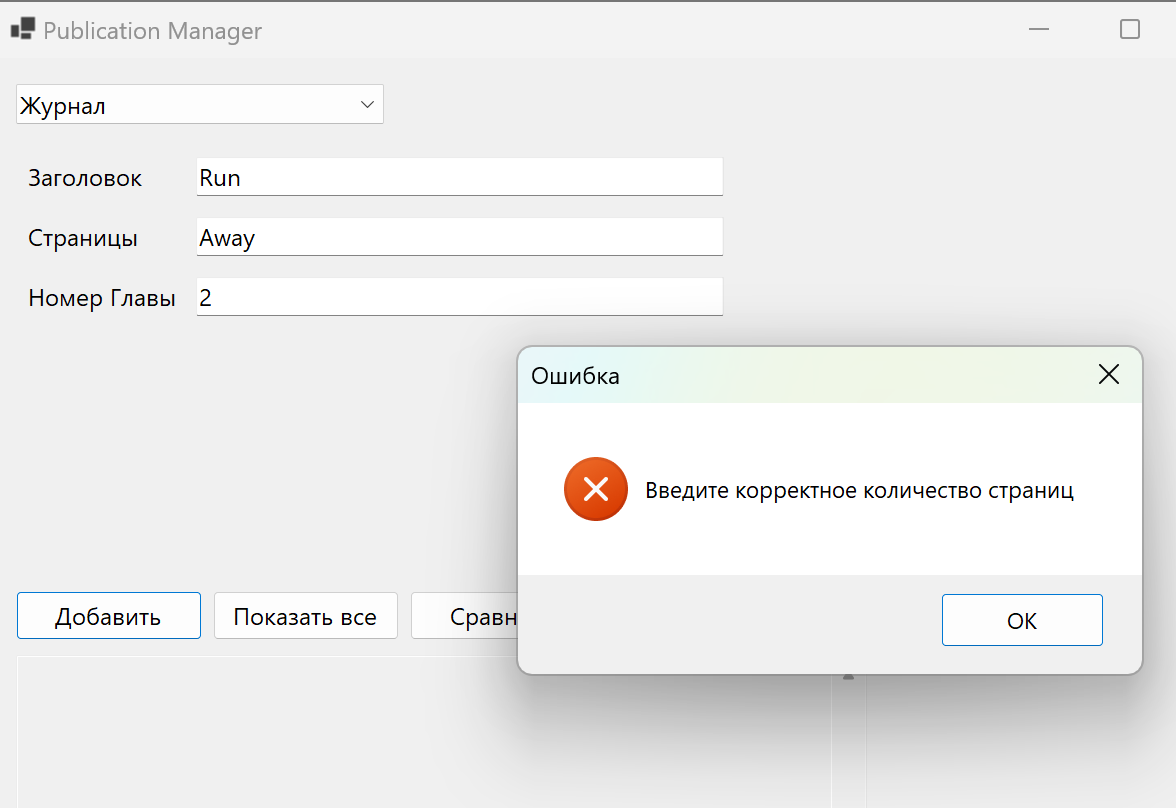


Рис. 2 – Предупреждения для ввода чисел

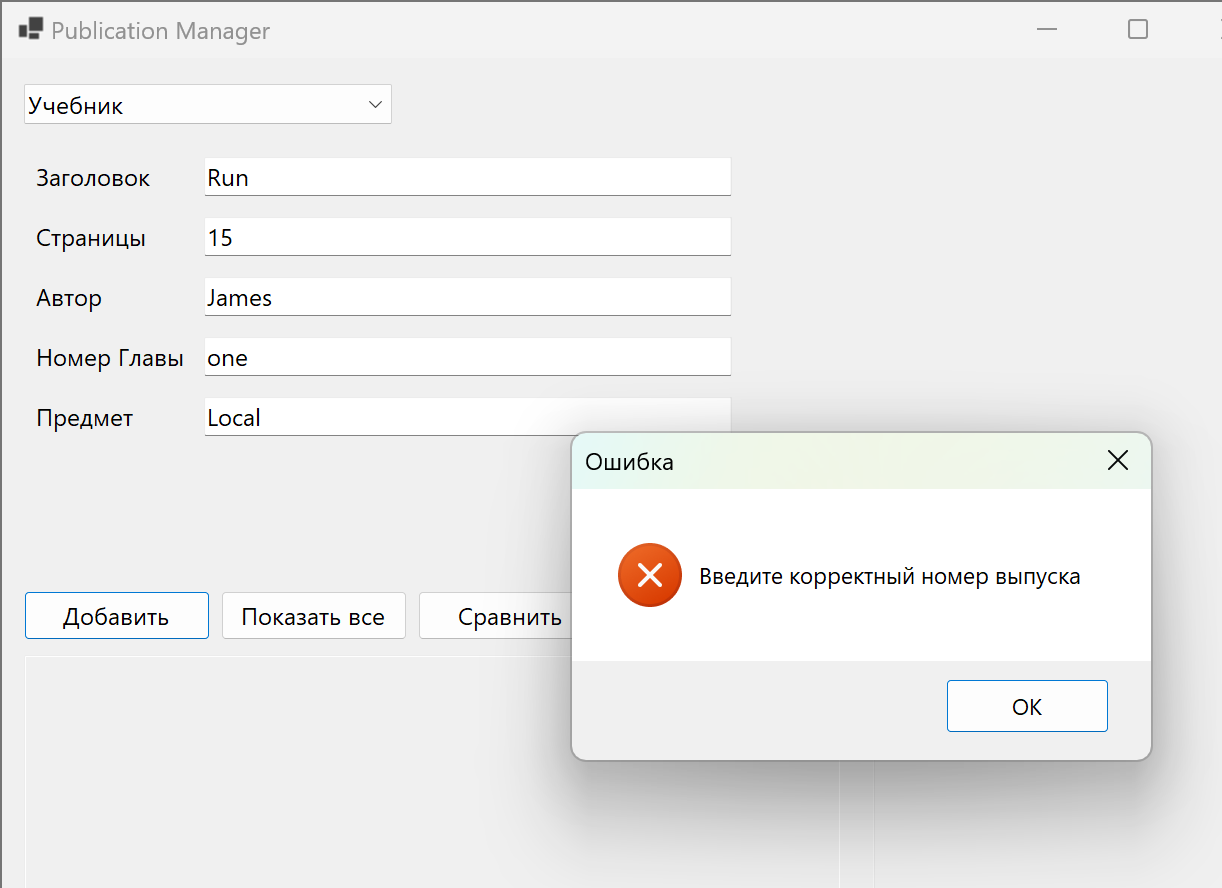


Рис. 3 – Предупреждения для ввода чисел

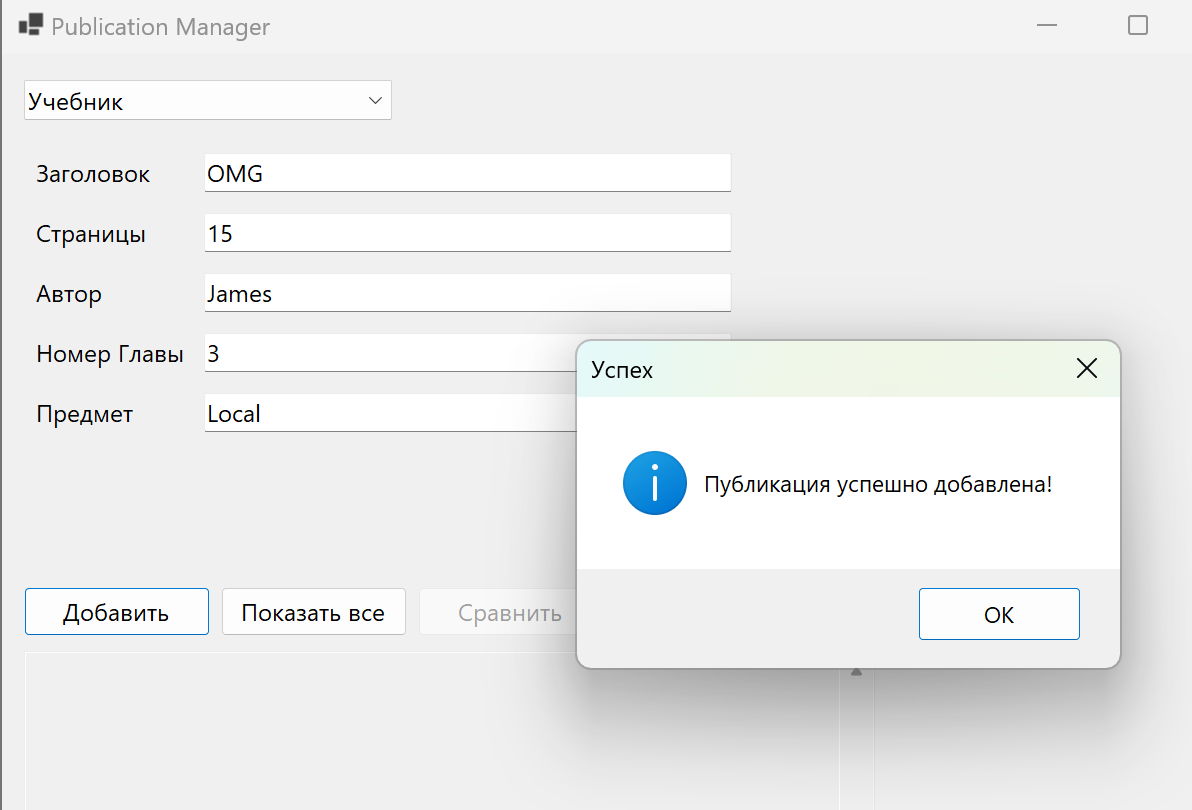


Рис. 4 – Ввод данных для класса “Учебник”

**3. Реализация и тестирование приложения**

3.1. Описание разработанной программы

Программа, разработанная на языке C#, позволяет создать иерархию классов. При создании иерархии классов используется алмаз смерти.

3.2. Тестирование программы

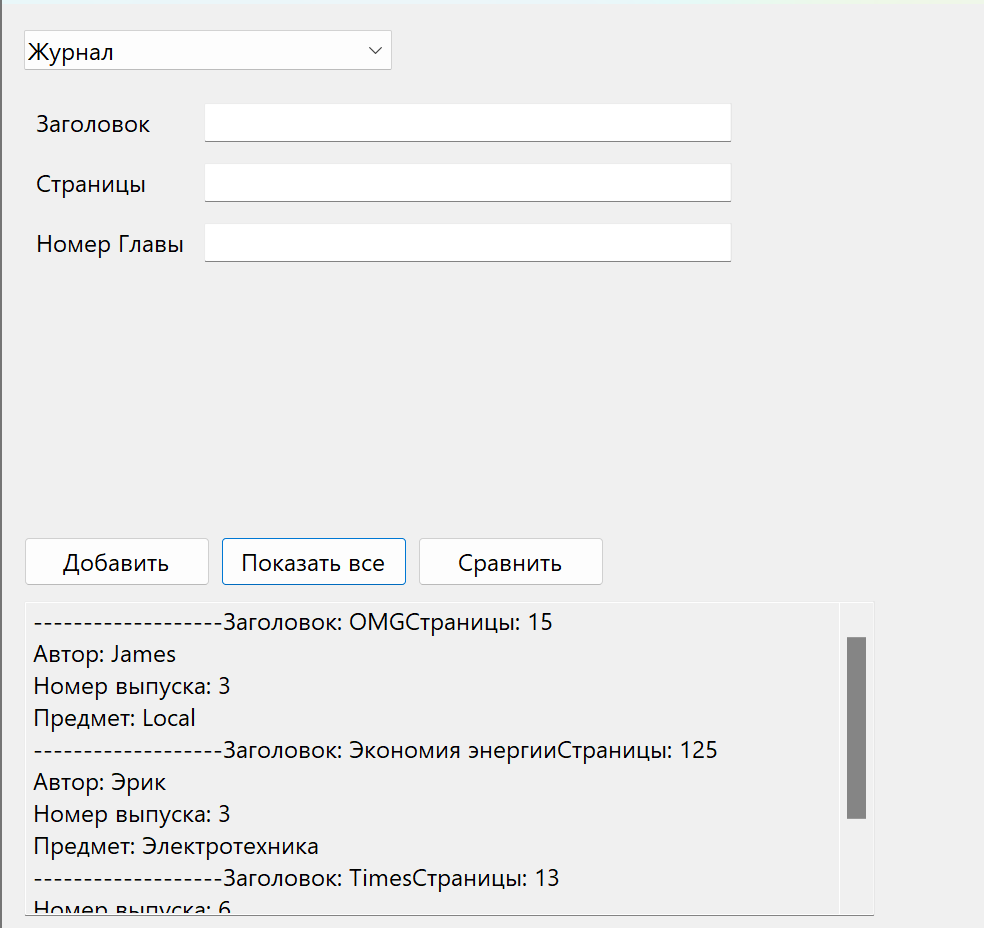
****

Рис. 5 – Просмотр введенных данных

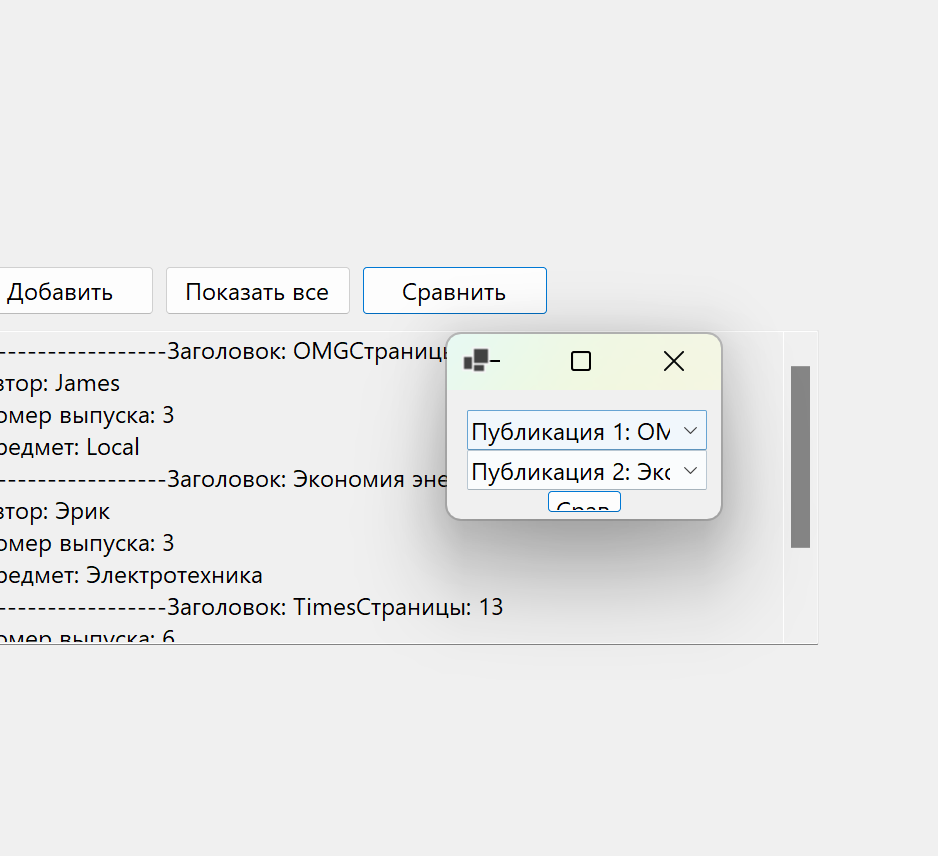


Рис. 6 – Сравнение 2х введенных данных

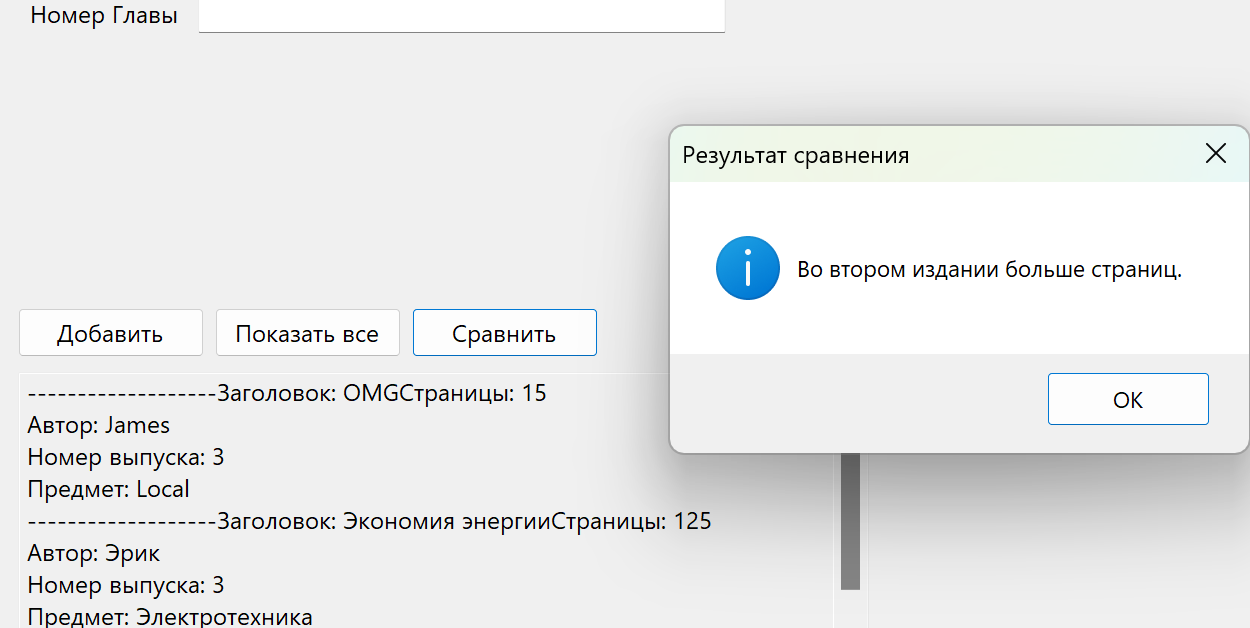


Рис. 7 – Вывод результаты сравнения

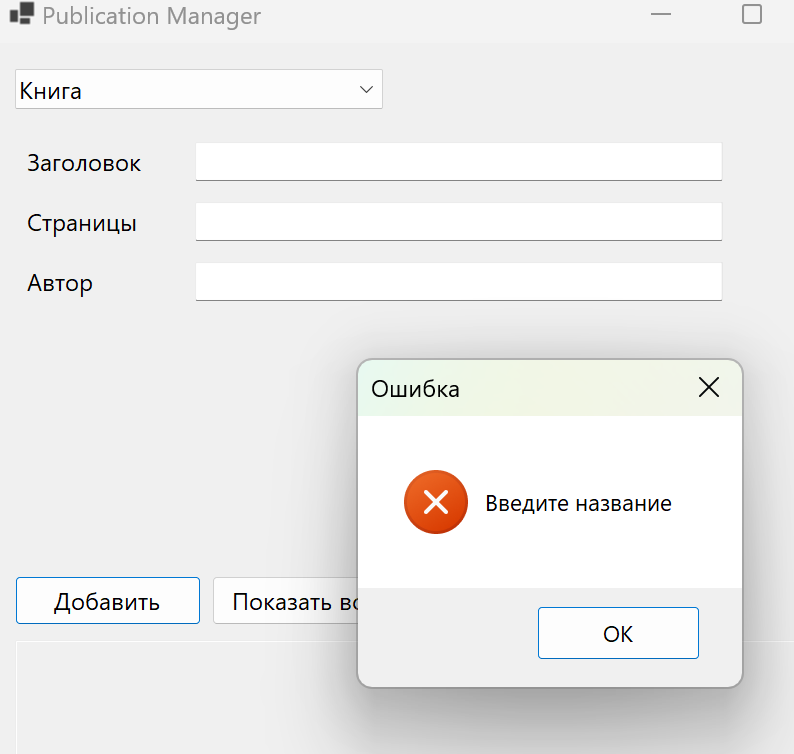


Рис. 8 – Попытка ввода данных класса “Book” с нулевыми данными

**Вывод**

В данной программе была создана иерархия классов. В языке программирования С# не возникает проблема множественного наследования, так как оно не реализуется в данном языке программирования. Для создания иерархии классов использовались интерфейсы. Языки С++ и С# не сильно различаются, поэтому сложность возникла только с наследованием. Так же было написано оконное приложение, именно его реализация заняла большую часть времени, так как необходимо было не только расположить элементы на форме, но и придать им характеристики (видимость, текст и тд). Помимо этого был обработан ряд исключений при использовании конструкции try catch.

**Приложение 1. Листинг программы.**

***Program.cs:***

namespace PublicationManager

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

// To customize application configuration such as set high DPI settings or default font,

// see https://aka.ms/applicationconfiguration.

ApplicationConfiguration.Initialize();

Application.Run(new Form1());

}

}

}

***Publication.cs:***

using System;

namespace PublicationManager

{

public abstract class AbstractPublication

{

public abstract string Title { get; }

public abstract void Display();

public abstract int GetPages();

public static void ComparePages(AbstractPublication pub1, AbstractPublication pub2)

{

if (pub1.GetPages() > pub2.GetPages())

Console.WriteLine("В первом издании больше страниц.");

else if (pub1.GetPages() < pub2.GetPages())

Console.WriteLine("Во втором издании больше страниц.");

else

Console.WriteLine("Обе публикации имют одинаковое колличество страниц.");

}

}

public interface IMagazine

{

int IssueNumber { get; set; }

void DisplayMagazineInfo();

}

public interface IBook

{

string Author { get; set; }

void DisplayBookInfo();

}

public class PrintPublication : AbstractPublication

{

private readonly string title;

private readonly int pages;

public override string Title => title;

public PrintPublication(string title, int pages)

{

this.title = title ?? throw new ArgumentNullException(nameof(title));

if (pages < 0)

throw new ArgumentException("Страницы не могут быть отрицательными");

this.pages = pages;

}

public override void Display()

{

Console.WriteLine($"Заголовок: {title}\nСтраницы: {pages}");

}

public override int GetPages()

{

return pages;

}

}

public class MagazineBase : PrintPublication, IMagazine

{

public int IssueNumber { get; set; }

public MagazineBase(string title, int pages, int issueNumber)

: base(title, pages)

{

IssueNumber = issueNumber;

}

public void DisplayMagazineInfo()

{

Console.WriteLine($"Номер выпуска: {IssueNumber}");

}

public override void Display()

{

base.Display();

DisplayMagazineInfo();

}

}

public class BookBase : PrintPublication, IBook

{

public string Author { get; set; }

public BookBase(string title, int pages, string author)

: base(title, pages)

{

Author = author ?? throw new ArgumentNullException(nameof(author));

}

public void DisplayBookInfo()

{

Console.WriteLine($"Автор: {Author}");

}

public override void Display()

{

base.Display();

DisplayBookInfo();

}

}

public class Textbook : PrintPublication, IMagazine, IBook

{

public int IssueNumber { get; set; }

public string Author { get; set; }

private readonly string subject;

public Textbook(string title, int pages, string author, int issueNumber, string subject)

: base(title, pages)

{

Author = author ?? throw new ArgumentNullException(nameof(author));

IssueNumber = issueNumber;

this.subject = subject ?? throw new ArgumentNullException(nameof(subject));

}

public void DisplayMagazineInfo()

{

Console.WriteLine($"Номер выпуска: {IssueNumber}");

}

public void DisplayBookInfo()

{

Console.WriteLine($"Автор: {Author}");

}

public override void Display()

{

base.Display();

DisplayBookInfo();

DisplayMagazineInfo();

Console.WriteLine($"Предмет: {subject}");

}

}

}

***Form1.cs:***

using System;

using System.Collections.Generic;

using System.Windows.Forms;

using System.Drawing;

namespace PublicationManager

{

public partial class Form1 : Form

{

private List<AbstractPublication> publications = new List<AbstractPublication>();

public Form1()

{

InitializeComponent();

SetupEventHandlers();

}

private void SetupEventHandlers()

{

publicationTypeCombo.SelectedIndexChanged += PublicationTypeCombo\_SelectedIndexChanged;

addButton.Click += AddButton\_Click;

displayAllButton.Click += DisplayAllButton\_Click;

compareButton.Click += CompareButton\_Click;

}

private void UpdateInputFields()

{

inputPanel!.Controls.Clear();

inputPanel.RowCount = 0;

AddField(titleLabel!, titleTextBox!);

AddField(pagesLabel!, pagesTextBox!);

if (publicationTypeCombo!.SelectedIndex >= 0)

{

switch (publicationTypeCombo!.SelectedItem?.ToString())

{

case "Журнал":

AddField(issueNumberLabel!, issueNumberTextBox!);

break;

case "Книга":

AddField(authorLabel!, authorTextBox!);

break;

case "Учебник":

AddField(authorLabel!, authorTextBox!);

AddField(issueNumberLabel!, issueNumberTextBox!);

AddField(subjectLabel!, subjectTextBox!);

break;

}

}

}

private void AddField(Label label, Control control)

{

int row = inputPanel!.RowCount++;

inputPanel.Controls.Add(label, 0, row);

inputPanel.Controls.Add(control, 1, row);

}

private void PublicationTypeCombo\_SelectedIndexChanged(object? sender, EventArgs e)

{

UpdateInputFields();

}

private void AddButton\_Click(object? sender, EventArgs e)

{

try

{

string title = titleTextBox!.Text;

if (string.IsNullOrWhiteSpace(title))

throw new ArgumentException("Введите название");

if (!int.TryParse(pagesTextBox!.Text, out int pages) || pages <= 0)

throw new ArgumentException("Введите корректное количество страниц");

string selectedType = publicationTypeCombo!.SelectedItem?.ToString() ?? string.Empty;

AbstractPublication? publication = null;

switch (selectedType)

{

case "Журнал":

if (!int.TryParse(issueNumberTextBox!.Text, out int issueNumber))

throw new ArgumentException("Введите корректный номер выпуска");

publication = new MagazineBase(titleTextBox!.Text, pages, issueNumber);

break;

case "Книга":

if (string.IsNullOrWhiteSpace(authorTextBox!.Text))

throw new ArgumentException("Введите автора");

publication = new BookBase(titleTextBox!.Text, pages, authorTextBox!.Text);

break;

case "Учебник":

if (string.IsNullOrWhiteSpace(authorTextBox!.Text))

throw new ArgumentException("Введите автора");

if (!int.TryParse(issueNumberTextBox!.Text, out issueNumber))

throw new ArgumentException("Введите корректный номер выпуска");

if (string.IsNullOrWhiteSpace(subjectTextBox!.Text))

throw new ArgumentException("Введите предмет");

publication = new Textbook(titleTextBox!.Text, pages, authorTextBox!.Text, issueNumber, subjectTextBox!.Text);

break;

}

if (publication != null)

{

publications.Add(publication);

UpdateCompareButton();

MessageBox.Show("Публикация успешно добавлена!", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearInputFields();

}

}

catch (Exception ex)

{

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

}

}

private void DisplayAllButton\_Click(object? sender, EventArgs e)

{

displayTextBox!.Clear();

if (publications.Count == 0)

{

displayTextBox.AppendText("Нет публикаций для отображения.");

return;

}

foreach (var publication in publications)

{

displayTextBox.AppendText("-------------------\n");

var oldOut = Console.Out;

using (var writer = new System.IO.StringWriter())

{

Console.SetOut(writer);

publication.Display();

Console.SetOut(oldOut);

displayTextBox.AppendText(writer.ToString() + "\n");

}

}

}

private void CompareButton\_Click(object? sender, EventArgs e)

{

if (publications.Count < 2)

{

MessageBox.Show("Необходимо добавить как минимум 2 публикации для сравнения.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

using (var compareForm = new Form())

{

compareForm.Text = "Сравнение публикаций";

compareForm.Size = new Size(300, 200);

compareForm.StartPosition = FormStartPosition.CenterParent;

var firstCombo = new ComboBox

{

DropDownStyle = ComboBoxStyle.DropDownList,

Location = new Point(20, 20),

Width = 240

};

var secondCombo = new ComboBox

{

DropDownStyle = ComboBoxStyle.DropDownList,

Location = new Point(20, 60),

Width = 240

};

for (int i = 0; i < publications.Count; i++)

{

string item = $"Публикация {i + 1}: {publications[i].Title}";

firstCombo.Items.Add(item);

secondCombo.Items.Add(item);

}

var compareBtn = new Button

{

Text = "Сравнить",

Location = new Point(100, 100),

DialogResult = DialogResult.OK

};

compareForm.Controls.AddRange(new Control[] { firstCombo, secondCombo, compareBtn });

compareForm.AcceptButton = compareBtn;

if (compareForm.ShowDialog() == DialogResult.OK &&

firstCombo.SelectedIndex != -1 &&

secondCombo.SelectedIndex != -1)

{

var pub1 = publications[firstCombo.SelectedIndex];

var pub2 = publications[secondCombo.SelectedIndex];

using (var writer = new System.IO.StringWriter())

{

var oldOut = Console.Out;

Console.SetOut(writer);

AbstractPublication.ComparePages(pub1, pub2);

Console.SetOut(oldOut);

MessageBox.Show(writer.ToString(), "Результат сравнения", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

}

private void UpdateCompareButton()

{

compareButton!.Enabled = publications.Count >= 2;

}

private void ClearInputFields()

{

titleTextBox!.Clear();

pagesTextBox!.Clear();

authorTextBox!.Clear();

issueNumberTextBox!.Clear();

subjectTextBox!.Clear();

}

}

}

***Form1.Designer.cs:***

namespace PublicationManager;

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

// Control declarations

private System.Windows.Forms.ComboBox publicationTypeCombo;

private System.Windows.Forms.Button addButton;

private System.Windows.Forms.Button displayAllButton;

private System.Windows.Forms.Button compareButton;

private System.Windows.Forms.TableLayoutPanel inputPanel;

private System.Windows.Forms.Label titleLabel;

private System.Windows.Forms.TextBox titleTextBox;

private System.Windows.Forms.Label pagesLabel;

private System.Windows.Forms.TextBox pagesTextBox;

private System.Windows.Forms.Label issueNumberLabel;

private System.Windows.Forms.TextBox issueNumberTextBox;

private System.Windows.Forms.Label authorLabel;

private System.Windows.Forms.TextBox authorTextBox;

private System.Windows.Forms.Label subjectLabel;

private System.Windows.Forms.TextBox subjectTextBox;

private System.Windows.Forms.TextBox displayTextBox;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

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

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

publicationTypeCombo = new ComboBox();

addButton = new Button();

displayAllButton = new Button();

compareButton = new Button();

inputPanel = new TableLayoutPanel();

titleLabel = new Label();

titleTextBox = new TextBox();

pagesLabel = new Label();

pagesTextBox = new TextBox();

issueNumberLabel = new Label();

issueNumberTextBox = new TextBox();

authorLabel = new Label();

authorTextBox = new TextBox();

subjectTextBox = new TextBox();

subjectLabel = new Label();

displayTextBox = new TextBox();

inputPanel.SuspendLayout();

SuspendLayout();

//

// publicationTypeCombo

//

publicationTypeCombo.DropDownStyle = ComboBoxStyle.DropDownList;

publicationTypeCombo.FormattingEnabled = true;

publicationTypeCombo.Items.AddRange(new object[] { "Журнал", "Книга", "Учебник" });

publicationTypeCombo.Location = new Point(22, 26);

publicationTypeCombo.Margin = new Padding(6);

publicationTypeCombo.Name = "publicationTypeCombo";

publicationTypeCombo.Size = new Size(368, 40);

publicationTypeCombo.TabIndex = 0;

//

// addButton

//

addButton.Location = new Point(22, 533);

addButton.Margin = new Padding(6);

addButton.Name = "addButton";

addButton.Size = new Size(186, 49);

addButton.TabIndex = 3;

addButton.Text = "Добавить";

//

// displayAllButton

//

displayAllButton.Location = new Point(219, 533);

displayAllButton.Margin = new Padding(6);

displayAllButton.Name = "displayAllButton";

displayAllButton.Size = new Size(186, 49);

displayAllButton.TabIndex = 4;

displayAllButton.Text = "Показать все";

//

// compareButton

//

compareButton.Location = new Point(416, 533);

compareButton.Margin = new Padding(6);

compareButton.Name = "compareButton";

compareButton.Size = new Size(186, 49);

compareButton.TabIndex = 5;

compareButton.Text = "Сравнить";

//

// inputPanel

//

inputPanel.AutoSize = true;

inputPanel.ColumnCount = 2;

inputPanel.ColumnStyles.Add(new ColumnStyle());

inputPanel.ColumnStyles.Add(new ColumnStyle(SizeType.Percent, 100F));

inputPanel.Controls.Add(titleLabel, 0, 0);

inputPanel.Controls.Add(titleTextBox, 1, 0);

inputPanel.Controls.Add(pagesLabel, 0, 1);

inputPanel.Controls.Add(pagesTextBox, 1, 1);

inputPanel.Controls.Add(issueNumberLabel, 0, 2);

inputPanel.Controls.Add(issueNumberTextBox, 1, 2);

inputPanel.Controls.Add(authorLabel, 0, 3);

inputPanel.Controls.Add(authorTextBox, 1, 3);

inputPanel.Controls.Add(subjectTextBox, 1, 4);

inputPanel.Controls.Add(subjectLabel, 0, 4);

inputPanel.Location = new Point(22, 87);

inputPanel.Margin = new Padding(6);

inputPanel.Name = "inputPanel";

inputPanel.RowStyles.Add(new RowStyle(SizeType.Absolute, 60F));

inputPanel.RowStyles.Add(new RowStyle(SizeType.Absolute, 60F));

inputPanel.RowStyles.Add(new RowStyle(SizeType.Absolute, 60F));

inputPanel.RowStyles.Add(new RowStyle(SizeType.Absolute, 60F));

inputPanel.RowStyles.Add(new RowStyle(SizeType.Absolute, 60F));

inputPanel.Size = new Size(854, 427);

inputPanel.TabIndex = 1;

//

// titleLabel

//

titleLabel.Location = new Point(6, 15);

titleLabel.Margin = new Padding(6, 15, 6, 0);

titleLabel.Name = "titleLabel";

titleLabel.Size = new Size(162, 36);

titleLabel.TabIndex = 0;

titleLabel.Text = "Заголовок";

//

// titleTextBox

//

titleTextBox.Location = new Point(180, 12);

titleTextBox.Margin = new Padding(6, 12, 6, 0);

titleTextBox.Name = "titleTextBox";

titleTextBox.Size = new Size(528, 39);

titleTextBox.TabIndex = 1;

//

// pagesLabel

//

pagesLabel.Location = new Point(6, 75);

pagesLabel.Margin = new Padding(6, 15, 6, 0);

pagesLabel.Name = "pagesLabel";

pagesLabel.Size = new Size(124, 36);

pagesLabel.TabIndex = 2;

pagesLabel.Text = "Страницы";

//

// pagesTextBox

//

pagesTextBox.Location = new Point(180, 72);

pagesTextBox.Margin = new Padding(6, 12, 6, 0);

pagesTextBox.Name = "pagesTextBox";

pagesTextBox.Size = new Size(528, 39);

pagesTextBox.TabIndex = 3;

//

// issueNumberLabel

//

issueNumberLabel.Location = new Point(6, 135);

issueNumberLabel.Margin = new Padding(6, 15, 6, 0);

issueNumberLabel.Name = "issueNumberLabel";

issueNumberLabel.Size = new Size(162, 36);

issueNumberLabel.TabIndex = 4;

issueNumberLabel.Text = "Номер Главы";

//

// issueNumberTextBox

//

issueNumberTextBox.Location = new Point(180, 132);

issueNumberTextBox.Margin = new Padding(6, 12, 6, 0);

issueNumberTextBox.Name = "issueNumberTextBox";

issueNumberTextBox.Size = new Size(528, 39);

issueNumberTextBox.TabIndex = 5;

//

// authorLabel

//

authorLabel.Location = new Point(6, 195);

authorLabel.Margin = new Padding(6, 15, 6, 0);

authorLabel.Name = "authorLabel";

authorLabel.Size = new Size(148, 36);

authorLabel.TabIndex = 6;

authorLabel.Text = "Автор";

//

// authorTextBox

//

authorTextBox.Location = new Point(180, 192);

authorTextBox.Margin = new Padding(6, 12, 6, 0);

authorTextBox.Name = "authorTextBox";

authorTextBox.Size = new Size(528, 39);

authorTextBox.TabIndex = 7;

//

// subjectTextBox

//

subjectTextBox.Location = new Point(180, 252);

subjectTextBox.Margin = new Padding(6, 12, 6, 0);

subjectTextBox.Name = "subjectTextBox";

subjectTextBox.Size = new Size(528, 39);

subjectTextBox.TabIndex = 9;

//

// subjectLabel

//

subjectLabel.Location = new Point(6, 255);

subjectLabel.Margin = new Padding(6, 15, 6, 0);

subjectLabel.Name = "subjectLabel";

subjectLabel.Size = new Size(136, 36);

subjectLabel.TabIndex = 8;

subjectLabel.Text = "Предмет";

//

// displayTextBox

//

displayTextBox.Location = new Point(22, 597);

displayTextBox.Margin = new Padding(6);

displayTextBox.Multiline = true;

displayTextBox.Name = "displayTextBox";

displayTextBox.ReadOnly = true;

displayTextBox.ScrollBars = ScrollBars.Vertical;

displayTextBox.Size = new Size(851, 315);

displayTextBox.TabIndex = 2;

//

// Form1

//

AutoScaleDimensions = new SizeF(13F, 32F);

AutoScaleMode = AutoScaleMode.Font;

ClientSize = new Size(899, 943);

Controls.Add(publicationTypeCombo);

Controls.Add(inputPanel);

Controls.Add(displayTextBox);

Controls.Add(addButton);

Controls.Add(displayAllButton);

Controls.Add(compareButton);

Margin = new Padding(6);

Name = "Form1";

Text = "Publication Manager";

inputPanel.ResumeLayout(false);

inputPanel.PerformLayout();

ResumeLayout(false);

PerformLayout();

}

#endregion

}