Министерство науки и высшего образования

Российской Федерации

Федеральное государственное бюджетное

образовательное

учреждение высшего образования

«Владимирский государственный

университет

имени Александра Григорьевича и Николая

Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

Отчёт по лабораторной работе №3

Тема: Основы объектно-ориентированного программирования

Вариант №7

Выполнил

студент ИСТ-120

Голубев Д. О.

Проверил

Данилов В. В.

Владимир 2021

Оглавление

[Цель работы 3](#_Toc84417404)

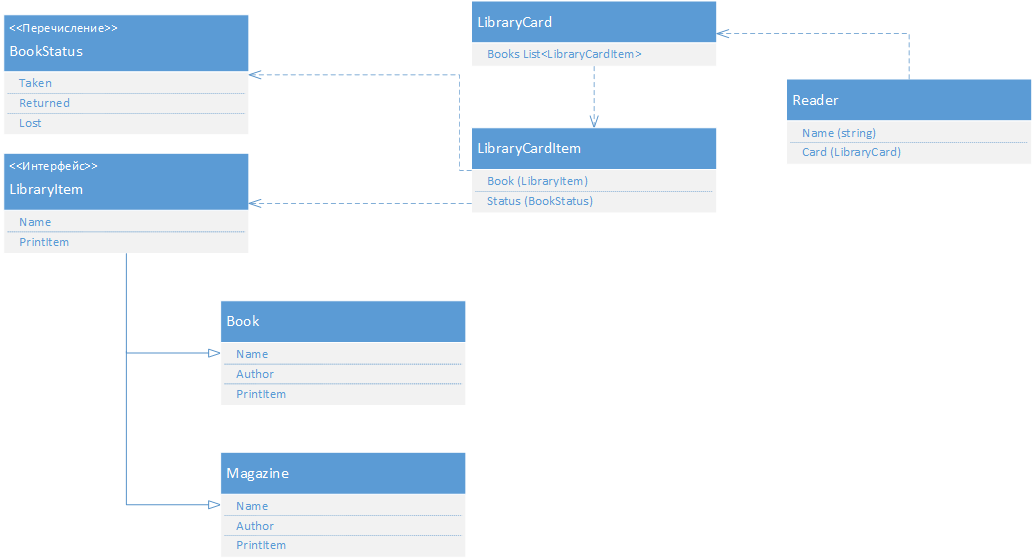
[ВЫПОЛНЕНИЕ РАБОТЫ 4](#_Toc84417405)

Цель работы

Изучить основные концепции и особенности объектно-ориентированного программирования (ООП) в языке c#. Создать приложение, иллюстрирующее основные принципы ООП.

ВЫПОЛНЕНИЕ РАБОТЫ

1 Анализ предметной области

Рисунок 1 - UML-диаграмма проекта

2 Исходный код

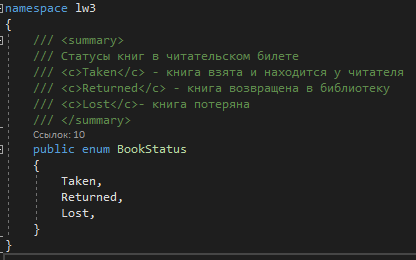


Рисунок 2 – BookStatus

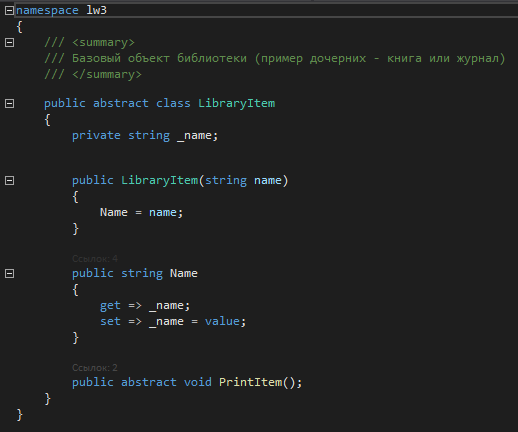


Рисунок 3 – LibraryItem

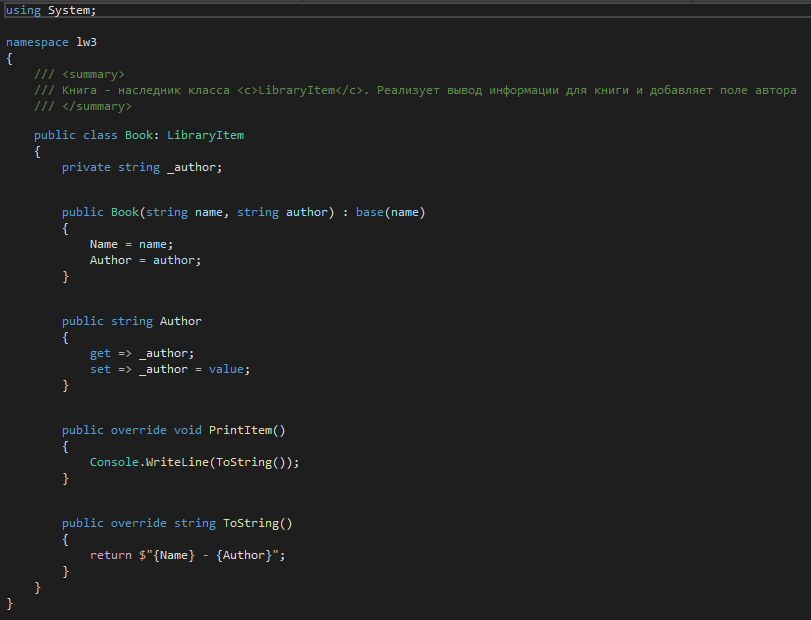


Рисунок 4 – Book

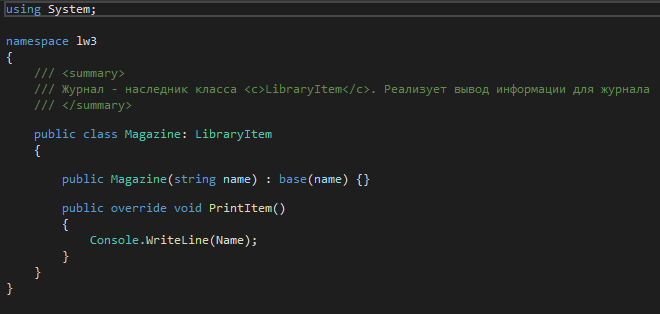


Рисунок 5 – Magazine

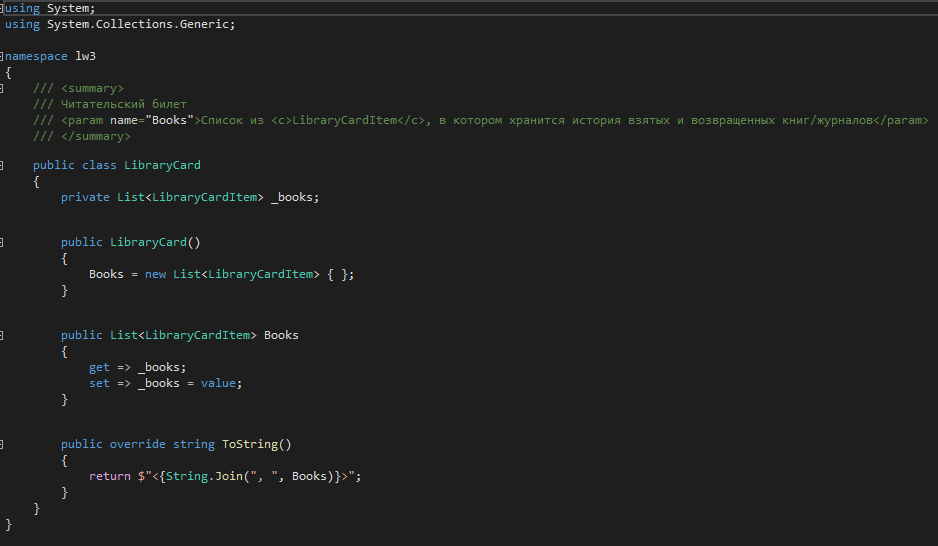


Рисунок 6 – LibraryCard



Рисунок 7 – LibraryBookItem

using System;

namespace lw3

{

/// <summary>

/// Читатель

/// <param name="Name">Имя читателя</param>

/// <param name="Card">Объект читательского билета (см. <see cref="LibraryCard"/>)</param>

/// </summary>

public class Reader

{

private string \_name;

private LibraryCard \_card;

public Reader(string name)

{

Name = name;

Card = new LibraryCard();

}

public string Name

{

get => \_name;

set => \_name = value;

}

public LibraryCard Card

{

get => \_card;

set => \_card = value;

}

/// <summary>

/// Добавляет новую книгу в читательский билет читателя и помечает её взятой

/// </summary>

/// <param name="book">Книга для добавления в читательский билет</param>

/// <returns>Новый элемент читательского билета</returns>

public LibraryCardItem TakeBook(LibraryItem book)

{

LibraryCardItem item = new LibraryCardItem(book, BookStatus.Taken);

Card.Books.Add(item);

return item;

}

/// <summary>

/// Возвращает книгу в библиотеку

/// </summary>

/// <param name="book">Книга, которую вернул читатель</param>

public void ReturnBook(LibraryItem book)

{

var item = \_findTakenBookInLibraryCard(book);

item.Status = BookStatus.Returned;

}

/// <summary>

/// Возвращает книгу в библиотеку

/// </summary>

/// <param name="book">Книга, которую вернул читатель</param>

public void ReturnBook(LibraryCardItem item)

{

item.Status = BookStatus.Returned;

}

/// <summary>

/// Утеря книги читателем

/// </summary>

/// <param name="book">Книга, которую потерял читатель</param>

public void LoseBook(LibraryItem book)

{

var item = \_findTakenBookInLibraryCard(book);

item.Status = BookStatus.Lost;

}

/// <summary>

/// Утеря книги читателем

/// </summary>

/// <param name="book">Книга, которую потерял читатель</param>

public void LoseBook(LibraryCardItem item)

{

item.Status = BookStatus.Lost;

}

/// <summary>

/// Ищет запись в читательском билете с указанной книгой, если она на руках

/// </summary>

/// <param name="book">Книга, которую нужно найти</param>

private LibraryCardItem \_findTakenBookInLibraryCard(LibraryItem book)

{

LibraryCardItem foundItem = null;

foreach (var item in Card.Books)

{

if (item.Book == book && item.Status == BookStatus.Taken)

{

foundItem = item;

break;

}

}

return foundItem;

}

/// <summary>

/// Печатает информацию о читателе

/// </summary>

public void PrintReaderInfo()

{

Console.WriteLine($"{Name} taken these books {Card}");

}

}

}

Листинг 1 – Reader



Рисунок 8 – Точка входа в программу

3 Контрольный запуск

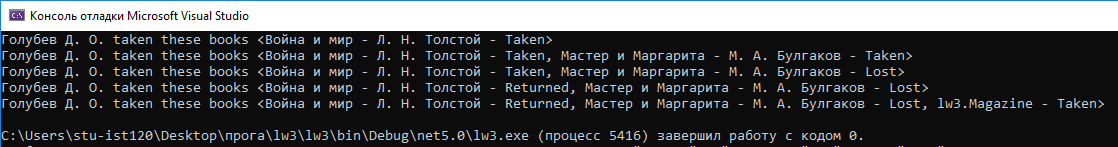


Рисунок 9 – Запуск программы

Вывод

В ходе выполнения работы были изучены основные концепции и особенности объектно-ориентированного программирования (ООП) в языке C# и создано приложение, иллюстрирующее основные принципы ООП.