

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**  
**ITMO University**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №9**

**По дисциплине** Объектно-ориентированное программирование

**Тема работы** Применение делегатов и событий

**Обучающийся** Буров Глеб Максимович

**Факультет** факультет инфокоммуникационных технологий

**Группа** K3223

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся**

\_\_\_\_\_  
(дата)

\_\_\_\_\_  
(подпись)

Буров Г.М.  
(Ф.И.О.)

**Руководитель**

\_\_\_\_\_  
(дата)

\_\_\_\_\_  
(подпись)

Иванов С.Е.  
(Ф.И.О.)

# СОДЕРЖАНИЕ

Стр.

<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>1   Ход работы .....</b>	<b>4</b>
1.1   Упражнение 1 .....	4
1.2   Упражнение 2 .....	5
1.3   Упражнение 3 .....	7
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>9</b>

## **ВВЕДЕНИЕ**

Целью данной лабораторной работы является использование делегатов и событий при реализации иерархии классов и приобретение навыков работы с ними.

# 1 Ход работы

## 1.1 Упражнение 1

В первом упражнении нужно было реализовать в классе Book из седьмой лабораторной работы возможность вызова метода через делегат.

Для этого в классе Book был объявлен делегат ProcessBookDelegate, принимающий в качестве аргументов объект класса Book. В проекте также был создан класс Operation, содержащий методы, с которыми связан делегат. Так, в классе Operation был описан статический метод PrintTitle, принимающий в качестве аргументов объект класса Book. Внутри метода PrintTitle у экземпляра вызывался метод Show.

В нашей идее предполагается, что вызов метода будет для тех книг, которые были возвращены в срок. Контроль за возвращением удобно проводить с помощью свойства, поэтому нужно было добавить в класс Book автоматическое свойство ReturnSrok. Также был добавлен метод ProcessPaperbackBooks, который находит возвращенные книги в срок и вызывает делегат для каждой книги (рис. 1.1).

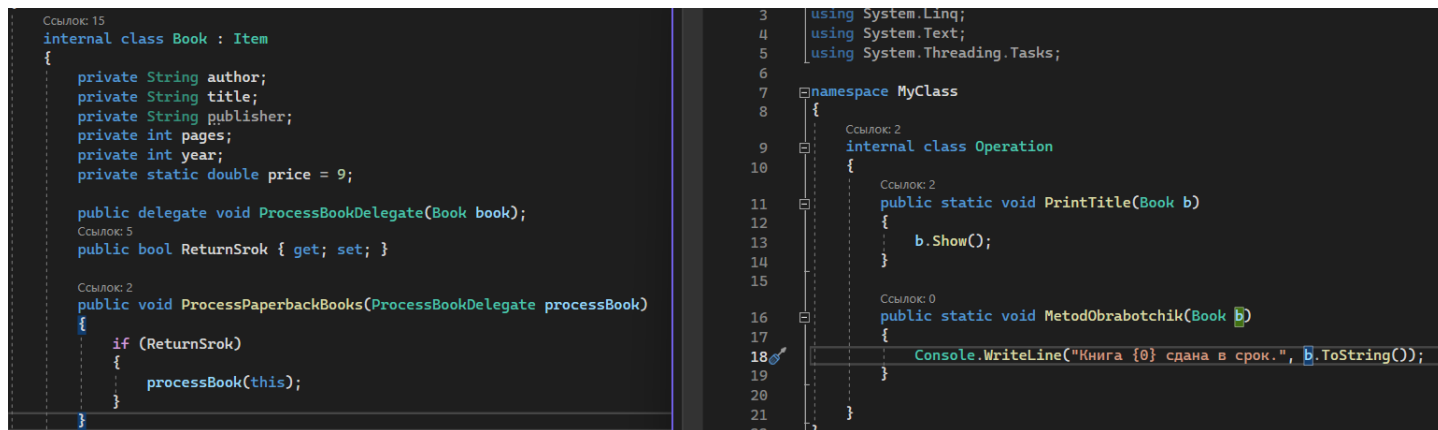


Рисунок 1.1 — Класс Operation и изменения в классе Book

В методе Main были созданы два новых экземпляра класса Book, и для них был вызван статический метод PrintTitle посредством делегата (рис. 1.2).

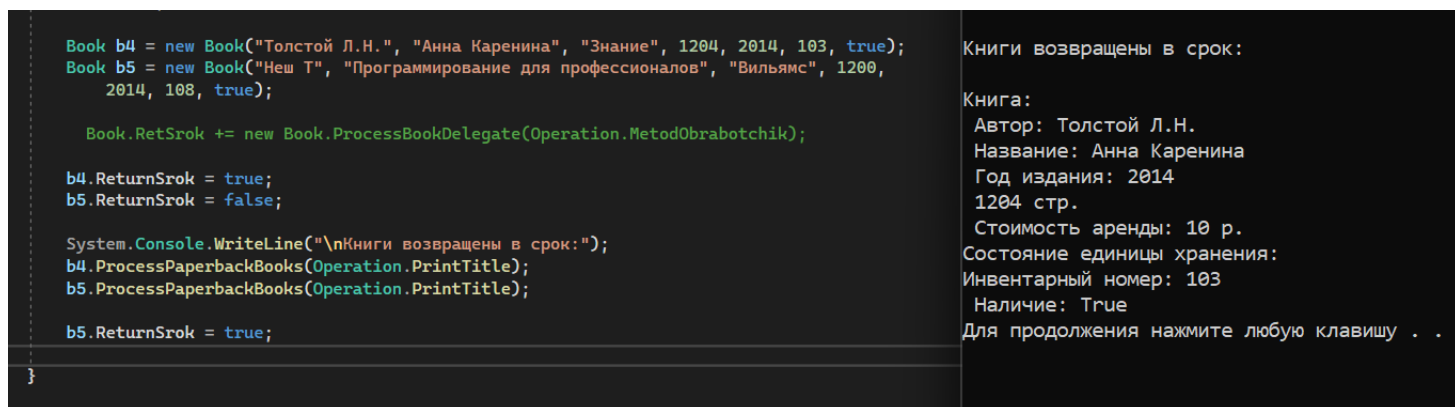


Рисунок 1.2 — Работа программы из упражнения 1

## 1.2 Упражнение 2

В рамках второго упражнения требовалось объявить событие «возвращение книги в срок», чтобы объекты класса Book смогли уведомлять объекты других классов о данном событии.

Для этого в классе Book было объявлено событие RetSrok на основе уже имеющегося делегата. Чтобы реализовать возможность наступления события при установке свойству ReturnSrok значения true, было удалено автоматическое свойство ReturnSrok, добавленное в прошлом упражнении и добавлено закрытое поле ReturnSrok и соответствующее ему свойство ReturnSrok. В сет-аксессоре нового свойства был добавлен код вызова события при наступлении требуемого условия (рис. 1.3).

```

internal class Book : Item
{
    private String author;
    private String title;
    private String publisher;
    private int pages;
    private int year;
    private static double price = 9;

    public delegate void ProcessBookDelegate(Book book);
    public static event ProcessBookDelegate RetSrok;
    private bool returnSrok = false;
    Ссылка: 6
    public bool ReturnSrok
    {
        get
        {
            return returnSrok;
        }
        set
        {
            returnSrok = value;
            if (ReturnSrok == true)
                RetSrok(this);
        }
    }
}

```

Рисунок 1.3 — Изменения в классе Book

В классе Book также был переопределен метод ToString для отображения объекта, а в класс Operation был добавлен метод MetodObrabotchik, который будет вызываться при наступлении события (рис. 1.1).

В методе класса Main был добавлен код, реализующий подписку объектов класса Book на событие. После была протестирована новая функциональность (рис. 1.4).

<pre> Book b4 = new Book("Толстой Л.Н.", "Анна Каренина", "Знание", 1204, 2014, 103); Book b5 = new Book("Неш Т", "Программирование для профессионалов", "Вильямс", 2014, 108, true);  Book.RetSrok += new Book.ProcessBookDelegate(Operation.MetodObrabotchik);  b4.ReturnSrok = true; b5.ReturnSrok = false;  System.Console.WriteLine("\nКниги возвращены в срок:"); b4.ProcessPaperbackBooks(Operation.PrintTitle); b5.ProcessPaperbackBooks(Operation.PrintTitle);  b5.ReturnSrok = true; } </pre>	<pre> Состояние единицы хранения: Инвентарный номер: 1235 Наличие: True Книга Анна Каренина, Толстой Л.Н. Инв.номер 103 сдана в срок.  Книги возвращены в срок:  Книга: Автор: Толстой Л.Н. Название: Анна Каренина Год издания: 2014 1204 стр. Стоимость аренды: 10 р. Состояние единицы хранения: Инвентарный номер: 103 Наличие: True Книга Программирование для профессионалов, Неш Т Инв.номер 108 сдана в срок. Для продолжения нажмите любую клавишу . . . </pre>
---	--

Рисунок 1.4 — Работа программы из упражнения 2

### 1.3 Упражнение 3

В последнем задании нужно было реализовать возникновение события «выпало максимальное количество очков» при броске игрального кубика в проекте IgralnayaKost из седьмой лабораторной работы.

Для этого в классе Gamer был объявлен делегат MaxNumberDelegate, принимающий в качестве аргумента объект класса Gamer. На основе этого делегата было объявлено событие «выпало максимальное количество очков» GotMaxNumber. В классе Gamer был также переопределен метод ToString, возвращающий имя игрока.

Был создан вспомогательный класс Operation, а в нём метод PrintInformation (аргумент — объект класса Gamer), выводящий информацию о том, что игроку выпало максимальное количество очков.

Метод SeansGame в классе Gamer был изменен так, чтобы при выбрасывании максимального количества очков наступало событие GotMaxNumber (рис. 1.5).

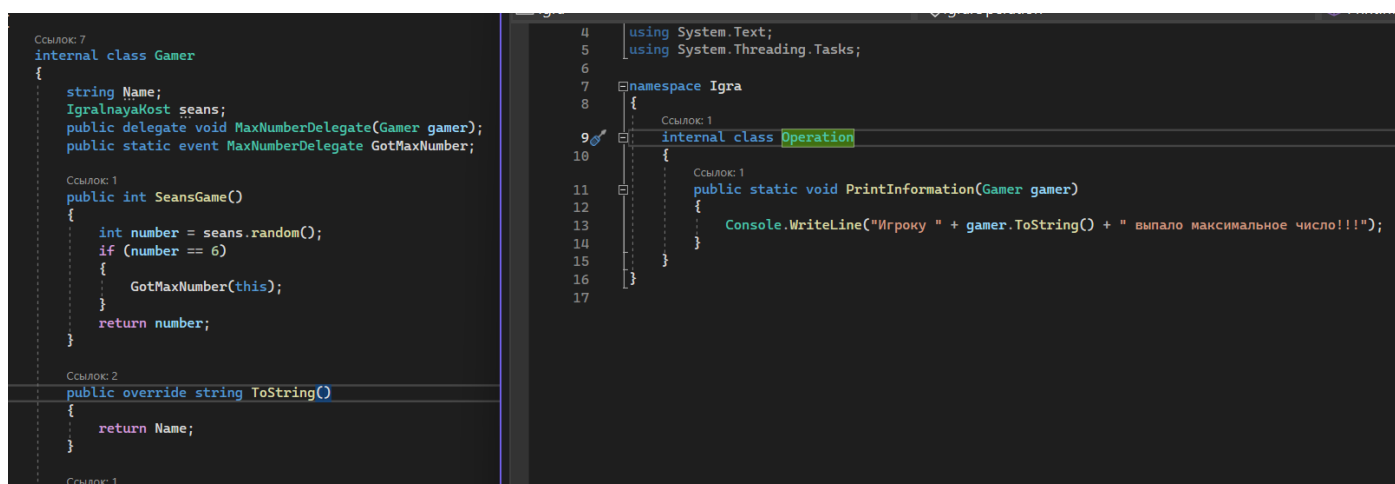


Рисунок 1.5 — Класс Operation и изменения в классе Gamer

Далее в классе Main был добавлен код, реализующий подписку объектов класса Gamer на событие. Пример работы программы показан на рис. 1.6.

```
static void Main(string[] args)
{
    Gamer g1 = new Gamer("Niko");
    Gamer.GotMaxNumber += new Gamer.MaxNumberDelegate(Operation.PrintInformation);
    for (int i = 1; i <= 6; i++)
    {
        Console.WriteLine("Выпало количество очков {0} для игрока {1}", g1.SeansGame(),
            g1.ToString());
    }
}
```

C:\Windows\system32\cmd.exe

Игроку Niko выпало максимальное число!!!  
Выпало количество очков 6 для игрока Niko  
Выпало количество очков 5 для игрока Niko  
Выпало количество очков 2 для игрока Niko  
Игроку Niko выпало максимальное число!!!  
Выпало количество очков 6 для игрока Niko  
Выпало количество очков 1 для игрока Niko  
Выпало количество очков 2 для игрока Niko  
Для продолжения нажмите любую клавишу . . .

Рисунок 1.6 — Работа программы из упражнения 3



## **ЗАКЛЮЧЕНИЕ**

В ходе этой лабораторной работы я ознакомился с делегатами и событиями, научился применять их в иерархии классов. Все упражнения были выполнены.