

Санкт-Петербургский Национальный Исследовательский  
Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №8**  
**Использование интерфейсов при реализации иерархии классов**

Выполнил  
Стафеев И.А.

Группа  
К3221

Проверил  
Иванов С.Е.

Санкт-Петербург,  
2024

## СОДЕРЖАНИЕ

	Стр.
<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>1 Упражнение 1.....</b>	<b>4</b>
<b>2 Упражнение 2.....</b>	<b>6</b>
<b>3 Упражнение 3.....</b>	<b>8</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>11</b>

## **ВВЕДЕНИЕ**

Цель работы: Использование интерфейсов при реализации иерархии классов как важного элемента объектно-ориентированного программирования и приобретение навыков реализации интерфейсов.

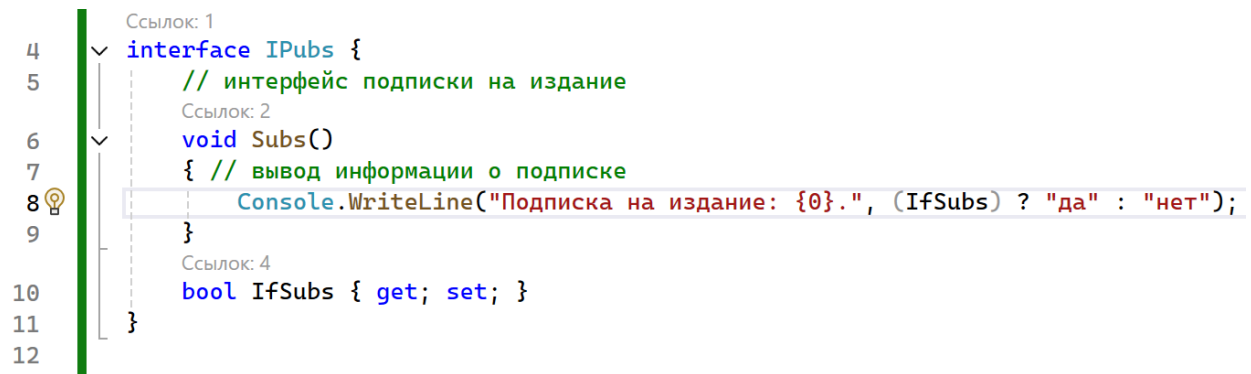
Для достижения цели необходимо выполнить следующие упражнения:

1. Создание и реализация интерфейса
2. Использование стандартных интерфейсов
3. Реализация прогрессии с помощью интерфейсов

## 1 Упражнение 1

Задача: добавить интерфейс для оформления подписки на печатные издания, дополни упражнения из предыдущей лабораторной работы.

Для добавления возможности оформления подписки был создан интерфейс IPubs, содержащий свойство IfSubs, указывающее, имеется ли подписка на издание, и метод Subs, выводящий информацию о подписке (1).

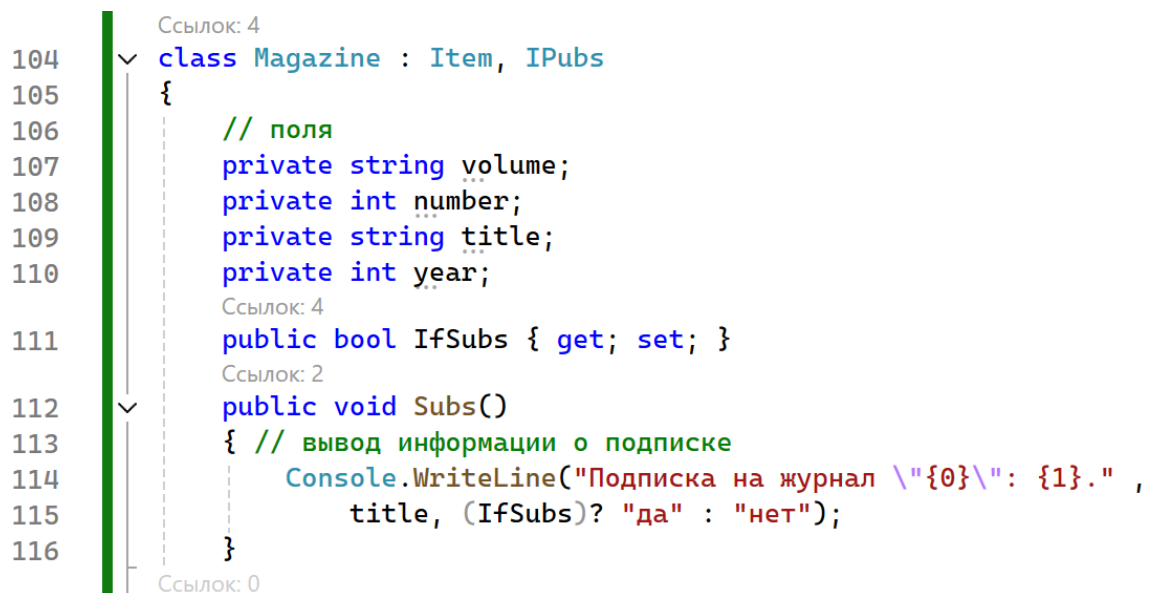


```
4  interface IPubs {  
5      // интерфейс подписки на издание  
6      void Subs()  
7      { // вывод информации о подписке  
8          Console.WriteLine("Подписка на издание: {0}.", (IfSubs) ? "да" : "нет");  
9      }  
10     bool IfSubs { get; set; }  
11 }  
12
```

Ссылки: 1, 2, 4

Рисунок 1 — Интерфейс IPubs

В рамках упражнения журнал будет считаться подписочным изданием, соответственно, реализующий его класс должен наследоваться от интерфейса и должен определять свойства и методы интерфейса (2).



```
104 class Magazine : Item, IPubs  
105 {  
106     // поля  
107     private string volume;  
108     private int number;  
109     private string title;  
110     private int year;  
111     public bool IfSubs { get; set; }  
112     public void Subs()  
113     { // вывод информации о подписке  
114         Console.WriteLine("Подписка на журнал \"{0}\": {1}.",  
115             title, (IfSubs)? "да" : "нет");  
116     }  
117 }
```

Ссылки: 4, 2, 0

Рисунок 2 — Измененный класс Magazine

Код метода Main для теста интерфейса представлен на рисунке 3.

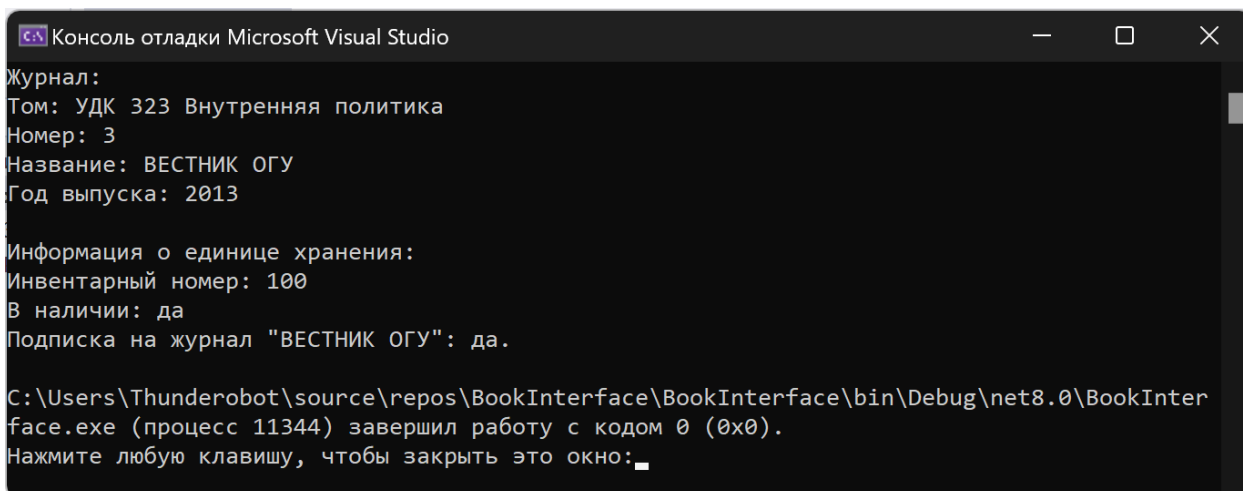
```

144 class Program
145 {
146     public static void Main(string[] args)
147     {
148         Magazine mag = new Magazine("УДК 323 Внутренняя политика", 3, "ВЕСТНИК ОГУ", 2013, 100, true);
149         mag.TakeItem();
150         mag.Return();
151         mag.Show();
152         mag.IfSubs = true;
153         mag.Subs();
154     }
155 }

```

Рисунок 3 — Метод Main

Результат выполнения программы показан на рисунке 4.



```

Консоль отладки Microsoft Visual Studio

Журнал:
Том: УДК 323 Внутренняя политика
Номер: 3
Название: ВЕСТНИК ОГУ
Год выпуска: 2013

Информация о единице хранения:
Инвентарный номер: 100
В наличии: да
Подписка на журнал "ВЕСТНИК ОГУ": да.

C:\Users\Thunderobot\source\repos\BookInterface\BookInterface\bin\Debug\net8.0\BookInter
face.exe (процесс 11344) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

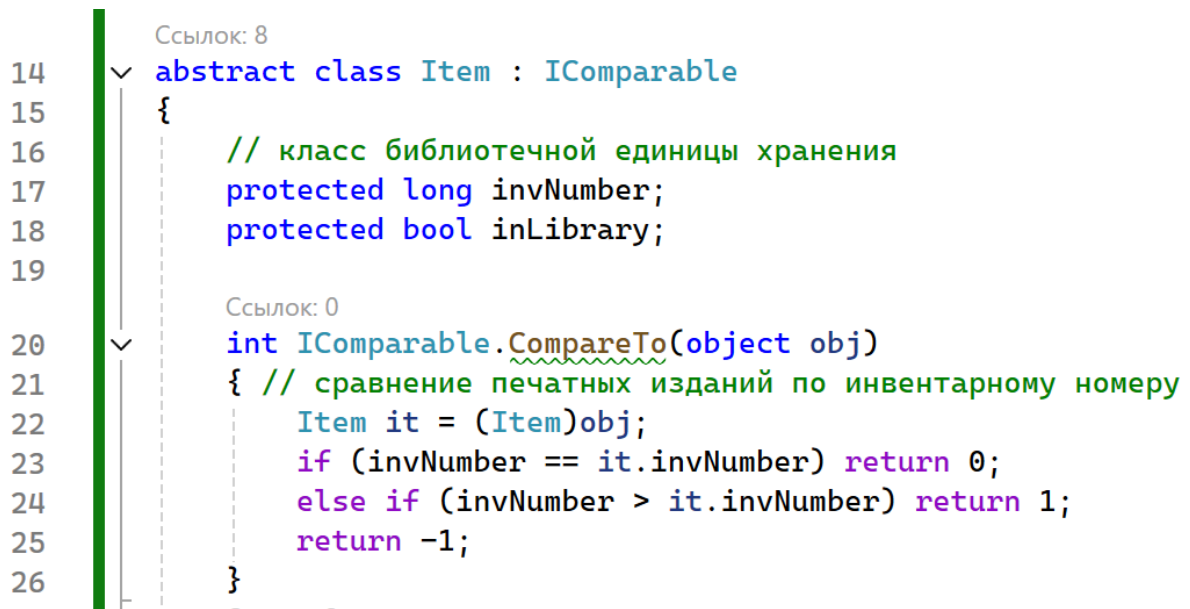
```

Рисунок 4 — Результат выполнения программы

## 2 Упражнение 2

Задача: применить стандартный интерфейс `Comparable` для реализации сравнения печатных изданий по идентификационному номеру.

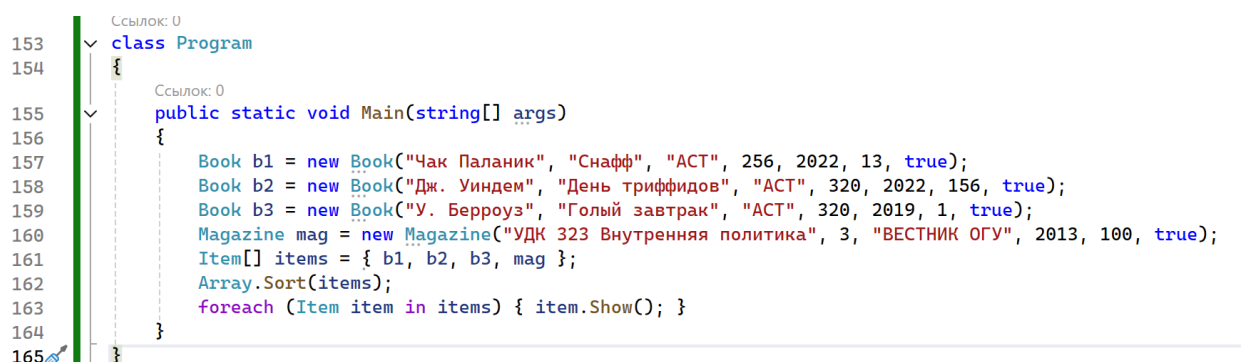
В список родителей класса `Item` был добавлен интерфейс `Comparable`, а также был создан метод `CompareTo`, сравнивающий печатные издания по инвентарному номеру (рис. 5).



```
14  abstract class Item : IComparable
15  {
16      // класс библиотечной единицы хранения
17      protected long invNumber;
18      protected bool inLibrary;
19
20      int IComparable.CompareTo(object obj)
21      { // сравнение печатных изданий по инвентарному номеру
22          Item it = (Item)obj;
23          if (invNumber == it.invNumber) return 0;
24          else if (invNumber > it.invNumber) return 1;
25          return -1;
26      }
```

Рисунок 5 — Добавление интерфейса к классу `Item`

Для теста в методе `Main` создается массив из 4 печатных изданий, который затем сортируется и выводится (6).



```
153  class Program
154  {
155      public static void Main(string[] args)
156      {
157          Book b1 = new Book("Чак Паланик", "Снафф", "АСТ", 256, 2022, 13, true);
158          Book b2 = new Book("Дж. Уиндем", "День триффидов", "АСТ", 320, 2022, 156, true);
159          Book b3 = new Book("У. Берроуз", "Голый завтрак", "АСТ", 320, 2019, 1, true);
160          Magazine mag = new Magazine("УДК 323 Внутренняя политика", 3, "ВЕСТНИК ОГУ", 2013, 100, true);
161          Item[] items = { b1, b2, b3, mag };
162          Array.Sort(items);
163          foreach (Item item in items) { item.Show(); }
164      }
165  }
```

Рисунок 6 — Метод `Main`

Результат выполнения программы показан на рисунке 7.

```
Консоль отладки Microsoft Visual Studio

Книга:
Название: Голый завтрак
Автор: У. Берроуз
Год издания: 2019
Количество страниц: 320
Стоимость аренды: 15 р. в сутки
Информация о единице хранения:
Инвентарный номер: 1
В наличии: да

Книга:
Название: Снафф
Автор: Чак Паланик
Год издания: 2022
Количество страниц: 256
Стоимость аренды: 15 р. в сутки
Информация о единице хранения:
Инвентарный номер: 13
В наличии: да

Журнал:
Том: УДК 323 Внутренняя политика
Номер: 3
Название: ВЕСТНИК ОГУ
Год выпуска: 2013

Информация о единице хранения:
Инвентарный номер: 100
В наличии: да

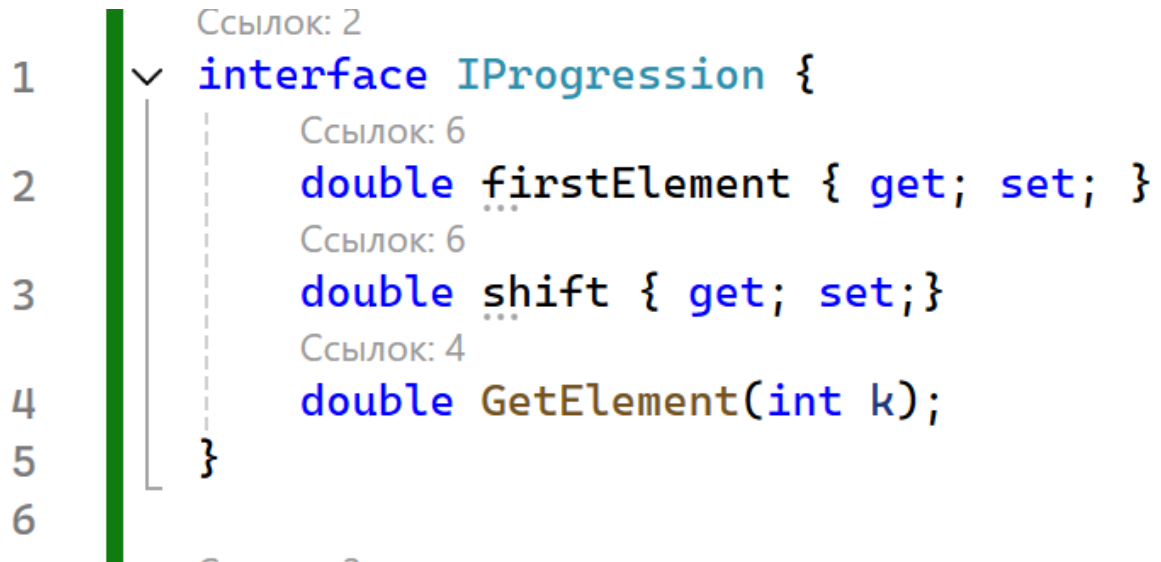
Книга:
Название: День триффидов
Автор: Дж. Уиндем
Год издания: 2022
Количество страниц: 320
Стоимость аренды: 15 р. в сутки
Информация о единице хранения:
Инвентарный номер: 156
В наличии: да
```

Рисунок 7 — Результат выполнения программы

### 3 Упражнение 3

Задача: необходимо переписать код из предыдущей лабораторной работы, реализующий прогрессии, с использованием интерфейса.

Код интерфейса IProgression показан на рисунке 8. В нем созданы два свойства - первый элемент прогрессии и изменение - и метод для получения  $k$ -го элемента.



```
1  interface IProgression {  
2      double firstElement { get; set; }  
3      double shift { get; set; }  
4      double GetElement(int k);  
5  }  
6
```

Рисунок 8 — Интерфейс IProgression

Классы арифметической и геометрической прогрессии наследуются от созданного интерфейса (9), в них реализуются свойства и метод интерфейса.



```

7      Ссылка: 2
      class ArithmeticProgression : IProgression {
8          Ссылка: 3
          public double firstElement { get; set; }
          Ссылка: 3
9          public double shift { get; set; }
          Ссылка: 3
10         public double GetElement(int k) {
11             return firstElement + (k - 1) * shift;
12         }
13     }
14
15     Ссылка: 2
16     class GeometricProgression : IProgression
17     {
18         Ссылка: 3
19         public double firstElement { get; set; }
20         Ссылка: 3
21         public double shift { get; set; }
22         Ссылка: 1
23         public double GetElement(int k)
24         {
25             return firstElement * Math.Pow(shift, k - 1);
26         }
27     }

```

Рисунок 9 — Классы ArithmeticProgression и GeometricProgression

В методе Main у пользователя запрашиваются данные для прогрессий, после чего выводится их  $k$ -й элемент (рис. 10).

```

26 class Program {
27     public static void Main(string[] args) {
28         Console.Write("a0=");
29         double a0 = double.Parse(Console.ReadLine());
30         Console.Write("d=");
31         double d = double.Parse(Console.ReadLine());
32         ArithmeticProgression arprog = new ArithmeticProgression();
33         arprog.shift = d; arprog.firstElement = a0;
34         Console.Write("k=");
35         int k = int.Parse(Console.ReadLine());
36         Console.WriteLine("k-й член арифметической прогрессии равен {0}", arprog.GetElement(k));
37
38         Console.Write("b0=");
39         double b0 = double.Parse(Console.ReadLine());
40         Console.Write("q=");
41         double q = double.Parse(Console.ReadLine());
42         GeometricProgression geoprog = new GeometricProgression();
43         geoprog.shift = q; geoprog.firstElement = b0;
44         Console.Write("k=");
45         k = int.Parse(Console.ReadLine());
46         Console.WriteLine("k-й член геометрической прогрессии равен {0}", arprog.GetElement(k));
47     }
48 }
49

```

Рисунок 10 — Метод Main

Пример выполнения программы показан на рисунке 11.

```

Консоль отладки Microsoft Visual Studio
a0=13
d=7
k=235
k-й член арифметической прогрессии равен 1651
b0=12
q=1,25
k=54
k-й член геометрической прогрессии равен 384

C:\Users\Thunderobot\source\repos\ProgressionInterface\ProgressionInterface\bin\Debug\net
8.0\ProgressionInterface.exe (процесс 412) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

```

Рисунок 11 — Пример выполнения программы

## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были выполнены все требуемые упражнения. Цель работы достигнута. Получены знания об интерфейсах в языке C# и приобретены навыки их реализации и использования.