

Министерство науки и высшего образования Российской Федерации

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ITMO University

ЛАБОРАТОРНАЯ РАБОТА №7

По дисциплине Объектно-ориентированное программирование

Тема работы Создание иерархии классов

Обучающийся Крестьянова Елизавета Федоровна

Факультет факультет инфокоммуникационных технологий

Группа К3223

Направление подготовки 11.03.02 Инфокоммуникационные технологии и
системы связи

Образовательная программа Программирование в
инфокоммуникационных системах

Обучающийся	_____	_____	<u>Крестьянова Е.Ф.</u>
	(дата)	(подпись)	(Ф.И.О.)

Руководитель	_____	_____	<u>Иванов С.Е.</u>
	(дата)	(подпись)	(Ф.И.О.)

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Упражнение 1. Реализация наследования классов.....	4
2 Упражнение 2. Использование конструкторов	6
3 Упражнение 3. Переопределение методов.....	14
4 Упражнение 4. Применение абстрактного класса и аб- страктных методов	18
5 Упражнение 5. Реализация модели включения	20
6 Упражнение 6. Реализация отношения ассоциации между классами	24
7 Упражнение 7. Реализация прогрессии.....	28
ЗАКЛЮЧЕНИЕ	33

ВВЕДЕНИЕ

В данном отчёте представлено выполнение лабораторной работы по дисциплине «Объектно-ориентированное программирование».

Цель данной работы - изучение наследования как важного элемента объектно-ориентированного программирования и приобретение навыков реализации иерархии классов.

1 УПРАЖНЕНИЕ 1. РЕАЛИЗАЦИЯ НАСЛЕДОВАНИЯ КЛАССОВ

Из предыдущей лабораторной работы был взят проект MyClass с классом Book. Был создан новый класс Item, его содержание можно увидеть на рисунке 1.1.

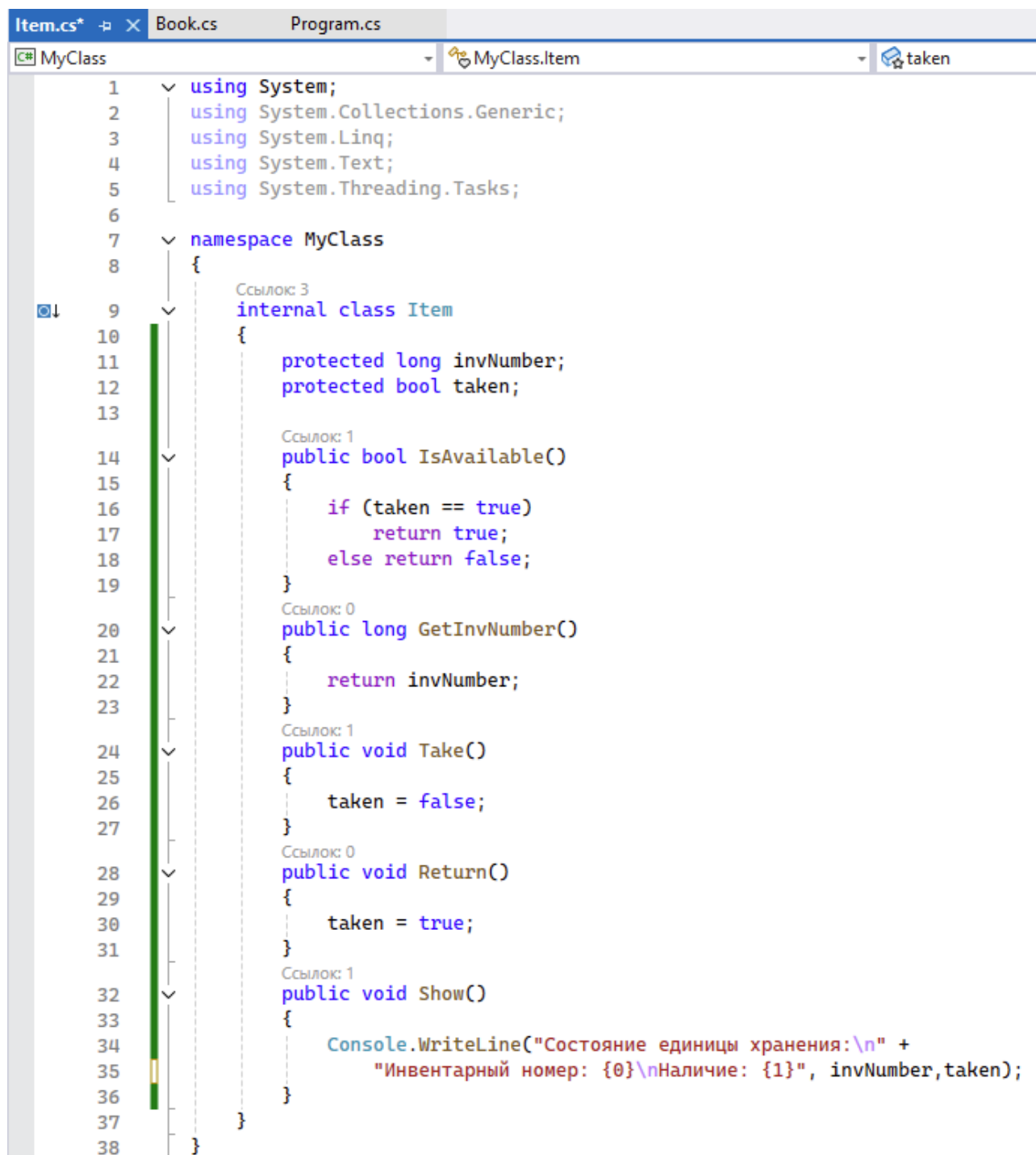


Рисунок 1.1 — Упр №1: Класс Item

В нем указываются две переменные, инвентарный номер и состояние объекта (взят ли он). Были определены 5 различных методов: вывод информации, вывод инвентарного номера, взятие и возврат предмета, проверка доступности.

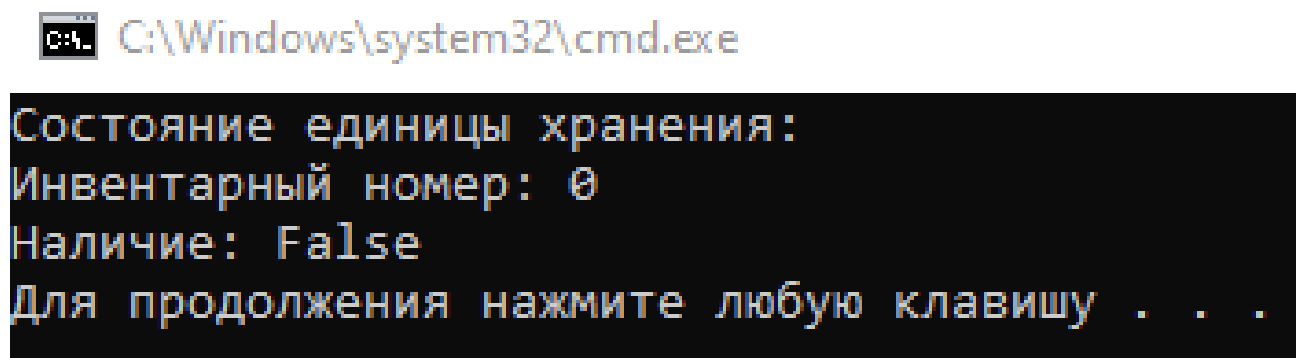
В классе Book был назначен базовый класс Item и создан новый метод TakeItem(): если книга доступна, пользователь может её взять. Так же для метода Show() было указано ключевое слово new. Нововведения можно увидеть на рисунке 1.2.

```
namespace MyClass
{
    Ссылка: 6
    internal class Book : Item
    {
        Ссылка: 0
        public void TakeItem()
        {
            if (this.IsAvailable()) this.Take();
        }

        Ссылка: 0
        new public void Show()
        {
            Console.WriteLine("\nКнига:\n Автор: {0}\n Название: {1}\n Год издания: {2}\n {3} стр
                               "\n Стоимость аренды: {4}", author, title, year, pages, Book.price);
        }
    }
}
```

Рисунок 1.2 — Упр №1: Изменения в классе Book

На рисунке 1.3 можно увидеть вывод программы.



```
C:\Windows\system32\cmd.exe

Состояние единицы хранения:
Инвентарный номер: 0
Наличие: False
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.3 — Упр №1: Вывод программы

2 УПРАЖНЕНИЕ 2. ИСПОЛЬЗОВАНИЕ КОНСТРУКТОРОВ

В этом упражнении было необходимо определить конструкторы в базовом и в производном классах и реализацию их выполнения с помощью ключевого слова `base`.

В `Item` были добавлены конструкторы по умолчанию и с параметром. Они представлены на рисунке 2.1.

```
namespace MyClass
{
    Ссылка: 7
    internal class Item
    {
        protected long invNumber;
        protected bool taken;

        Ссылка: 1
        public Item(long invNumber, bool taken)
        {
            this.invNumber = invNumber;
            this.taken = taken;
        }
    }
}
```

Рисунок 2.1 — Упр №2: Конструкторы класса `Item`

В классе `Book` был создан конструктор со ссылкой на базовый класс, и в методе `Show` был добавлен вызов метода `Show` у базового класса. Эти изменения можно увидеть на рисунке 2.2.

```
Ссылка: 1
public Book(String author, String title, String publisher,
    int pages, int year, long invNumber, bool taken) : base (invNumber, taken)
{
    this.author = author;
    this.title = title;
    this.publisher = publisher;
    this.pages = pages;
    this.year = year;
}

Ссылка: 1
new public void Show()
{
    Console.WriteLine("\nКнига:\n Автор: {0}\n Название: {1}\n Год издания: {2}\n {3} стр." +
        "\n Стоимость аренды: {4}", author, title, year, pages, Book.price);
    base.Show();
}
```

Рисунок 2.2 — Упр №2: Нововведения в классе Book

В методе Main к книге «Война и мир» были добавлены параметры для базового класса и вызваны методы выдачи книги и отображения данных о ней. Написанный код можно увидеть на рисунке 2.3.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClass
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             Item item1 = new Item();
14             item1.Show();
15
16             Book b2 = new Book("Толстой Л.Н.", "Война и мир",
17                               "Наука и жизнь", 1234, 2013, 101, true);
18             b2.TakeItem();
19             b2.Show();
20         }
21     }
22 }
23
24
```

Рисунок 2.3 — Упр №2: Создание новой книги и её взятие

Вывод информации об этой книге можно увидеть на рисунке 2.4.

 C:\Windows\system32\cmd.exe

```
Состояние единицы хранения:
  Инвентарный номер: 0
  Наличие: True

Книга:
  Автор: Толстой Л.Н.
  Название: Война и мир
  Год издания: 2013
  1234 стр.
  Стоимость аренды: 10
  Состояние единицы хранения:
    Инвентарный номер: 101
    Наличие: False
Для продолжения нажмите любую клавишу . . .
```

Рисунок 2.4 — Упр №2: Вывод информации о книге

Затем был добавлен класс Magazine для журналов. В нём указываются параметры том, номер, название и год выпуска, были добавлены конструкторы с параметрами и по умолчанию, добавлен метод показа всей информации. Весь класс можно увидеть на рисунке 2.5.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClass
8  {
9      internal class Magazine : Item
10     {
11         private String volume;
12         private int number;
13         private String title;
14         private int year;
15
16         public Magazine(string volume, int number, string title,
17             int year, long invNumber, bool taken) : base(invNumber, taken)
18         {
19             this.volume = volume;
20             this.number = number;
21             this.title = title;
22             this.year = year;
23         }
24         public Magazine() { }
25
26         new public void Show()
27         {
28             Console.WriteLine("\nЖурнал:\n Том: {0}\n Номер: {1}\n Название: " +
29                 "{2}\n Год выпуска: {3}", volume, number, title, year);
30             base.Show();
31         }
32     }
33 }
34
```

Рисунок 2.5 — Упр №2: Класс Magazine

На рисунке 2.6 представлено создание нового журнала "Земля и мы".

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClass
8  {
9      Ссылка: 0
10     internal class Program
11     {
12         Ссылка: 0
13         static void Main(string[] args)
14         {
15             Item item1 = new Item();
16             item1.Show();
17
18             Book b2 = new Book("Толстой Л.Н.", "Война и мир",
19                               "Наука и жизнь", 1234, 2013, 101, true);
20             b2.TakeItem();
21             b2.Show();
22
23             Magazine mag1 = new Magazine("О природе",
24                                          5, "Земля и мы", 2014, 1235, true);
25             mag1.Show();
26         }
27     }
28 }
29
```

Рисунок 2.6 — Упр №2: Создание нового журнала

На рисунке 2.7 показан вывод информации о журнале.

C:\Windows\system32\cmd.exe

```
Состояние единицы хранения:
  Инвентарный номер: 0
  Наличие: True

Книга:
  Автор: Толстой Л.Н.
  Название: Война и мир
  Год издания: 2013
  1234 стр.
  Стоимость аренды: 10
  Состояние единицы хранения:
    Инвентарный номер: 101
    Наличие: False

Журнал:
  Том: 0 природе
  Номер: 5
  Название: Земля и мы
  Год выпуска: 2014
  Состояние единицы хранения:
    Инвентарный номер: 1235
    Наличие: True
Для продолжения нажмите любую клавишу . . .
```

Рисунок 2.7 — Упр №2: Вывод информации о журнале

С помощью Visual Studio installer был включён флаг «Конструктор классов». В Visual Studio была создана диаграмма всех 4-ёх классов: Program, базового класса Item и его производных классов Book и Magazine. Эту диаграмму можно увидеть на рисунке 2.8.

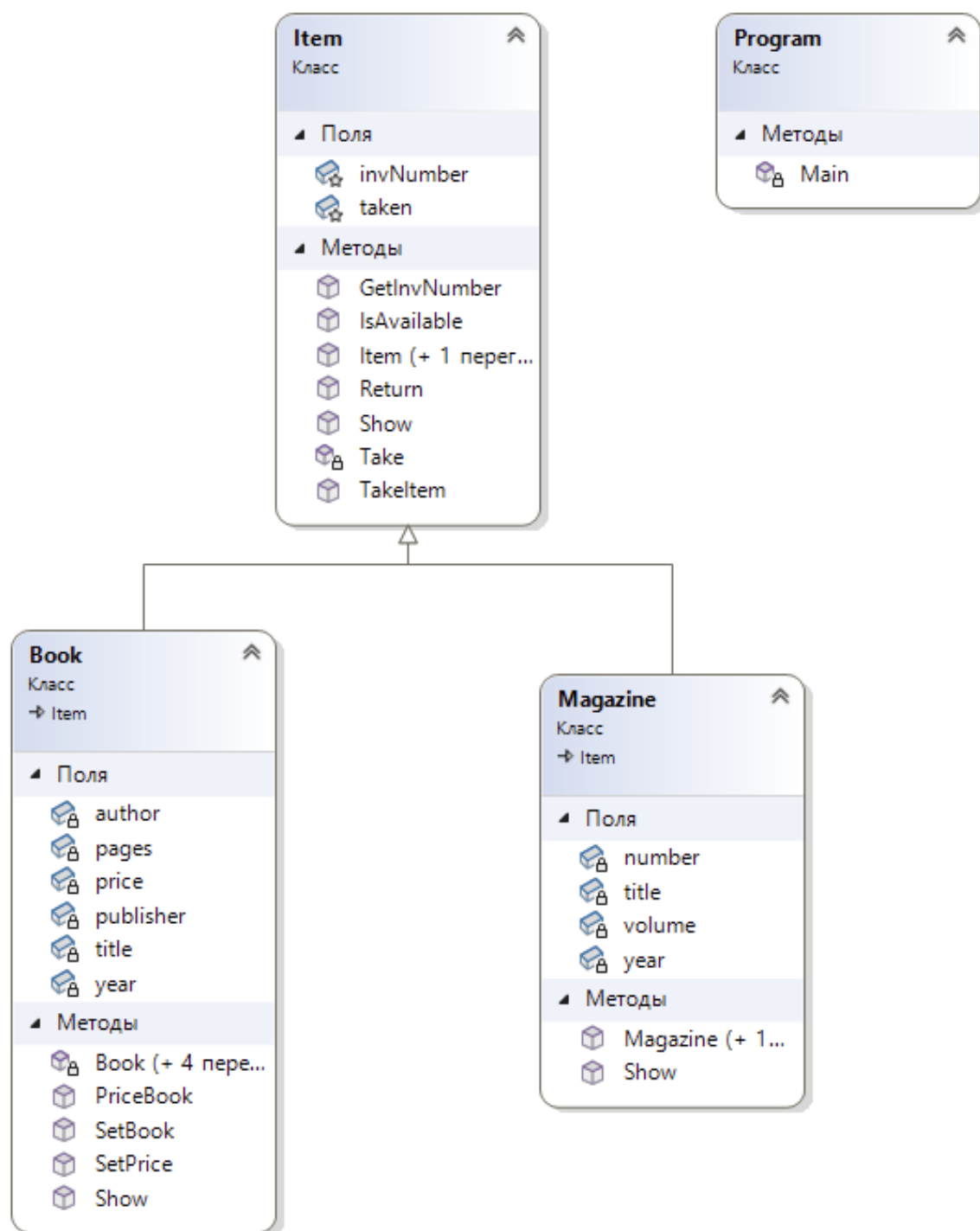


Рисунок 2.8 — Упр №2: Диаграмма классов

3 УПРАЖНЕНИЕ 3. ПЕРЕОПРЕДЕЛЕНИЕ МЕТОДОВ

В этом упражнении было необходимо реализовать механизм полиморфизма через виртуальные методы и их переопределение в производных классах.

В классе Item к методам Show() и Return() были добавлены ключевые слова virtual, чтобы объявить их виртуальными (Эти методы показаны на рисунке 3.1).

```
Ссылка: 2
public virtual void Return()
{
    taken = true;
}
Ссылка: 9
public virtual void Show()
{
    Console.WriteLine(" Состояние единицы хранения:\n" +
        " Инвентарный номер: {0}\n Наличие: {1}", invNumber, taken);
}
```

Рисунок 3.1 — Упр №3: Виртуализация методов Item

В класс Book было добавлено поле private bool returnSrok, устанавливающий, что книгу отдали в срок, и метод позволяющий это указать:

```
public void ReturnSrok()
{
    returnSrok = true;
}
```

В методе Show() ключевое слово new было изменено на ключевое слово override, позволяющий переопределять соответствующий виртуальный метод из базового класса Item. Был создан метод Return(), так же переопределяющий метод из Item. Их конструкцию можно увидеть на рисунке 3.2.

```

Ссылка: 7
public override void Show()
{
    Console.WriteLine("\nКнига:\n Автор: {0}\n Название: {1}\n Год издания: {2}\n {3} стр." +
        "\n Стоимость аренды: {4}", author, title, year, pages, Book.price);
    base.Show();
}

Ссылка: 1
public override void Return()
{
    if (returnSrok == true)
        taken = true;
    else
        taken = false;
}

```

Рисунок 3.2 — Упр №3: Переопределение методов в Book

Такие же методы Show() и Return() с ключевым словом override были оформлены и для класса Magazine, что показано на рисунке 3.3.

```

Ссылка: 7
public override void Show()
{
    Console.WriteLine("\nЖурнал:\n Том: {0}\n Номер: {1}\n Название: " +
        "{2}\n Год выпуска: {3}", volume, number, title, year);
    base.Show();
}

Ссылка: 1
public override void Return()
{
    taken = true;
}

```

Рисунок 3.3 — Упр №3: Переопределение методов в Magazine

В методе Main класса Program создаётся предмет it, которому присваиваются книга «Война и мир» и журнал «Земля и мы», они «забираются» и показывается информация о них. Присваивание этих объектов можно увидеть на рисунке 3.4.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClass
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             Item item1 = new Item();
14             item1.Show();
15
16             Book b2 = new Book("Толстой Л.Н.", "Война и мир",
17                               "Наука и жизнь", 1234, 2013, 101, true);
18             b2.TakeItem();
19             b2.Show();
20
21             Magazine mag1 = new Magazine("О природе",
22                                         5, "Земля и мы", 2014, 1235, true);
23             mag1.Show();
24
25             Console.WriteLine("\n Тестирование полиморфизма");
26             Item it;
27
28             it = b2;
29             it.TakeItem();
30             it.Show();
31
32             it = mag1;
33             it.TakeItem();
34             it.Show();
35         }
36     }
37 }
38
```

Рисунок 3.4 — Упр №3: Класс Program

Тем самым вызываются методы именно производных классов, а не базового, что видно на рисунке 3.5.

C:\. Выбрать C:\Windows\system32\cmd.exe

Состояние единицы хранения:

Инвентарный номер: 0

Наличие: True

Книга:

Автор: Толстой Л.Н.

Название: Война и мир

Год издания: 2013

1234 стр.

Стоимость аренды: 10

Состояние единицы хранения:

Инвентарный номер: 101

Наличие: False

Журнал:

Том: 0 природе

Номер: 5

Название: Земля и мы

Год выпуска: 2014

Состояние единицы хранения:

Инвентарный номер: 1235

Наличие: True

Тестирование полиморфизма

Книга:

Автор: Толстой Л.Н.

Название: Война и мир

Год издания: 2013

1234 стр.

Стоимость аренды: 10

Состояние единицы хранения:

Инвентарный номер: 101

Наличие: False

Журнал:

Том: 0 природе

Номер: 5

Название: Земля и мы

Год выпуска: 2014

Состояние единицы хранения:

Инвентарный номер: 1235

Наличие: False

Для продолжения нажмите любую клавишу . . .

Рисунок 3.5 — Упр №3: Вывод программы

4 УПРАЖНЕНИЕ 4. ПРИМЕНЕНИЕ АБСТРАКТНОГО КЛАССА И АБСТРАКТНЫХ МЕТОДОВ

В этом упражнении было необходимо сделать класс `Item` абстрактным, так как не имеет смысла создавать его объекты.

В строки `class Item, public void Return()` было добавлено по ключевому слову `abstract`, и из метода было удалено всё тело. Затем в классе `Program()` к книге и журналу был применён метод `Return()`. Вывод этой программы показан на рисунке 4.1. На нём видно, что журнал был возвращён, а книга нет: её поле `returnSrok` было ложным, что не позволило пройти условие. Значит переопределение методов явно сработало, и `Return()` вызывался у разных производных классов, не у базового.

C:\Windows\system32\cmd.exe

Название: Земля и мы
Год выпуска: 2014
Состояние единицы хранения:
Инвентарный номер: 1235
Наличие: True

Тестирование полиморфизма

Книга:

Автор: Толстой Л.Н.
Название: Война и мир
Год издания: 2013
1234 стр.
Стоимость аренды: 10
Состояние единицы хранения:
Инвентарный номер: 101
Наличие: False

Журнал:

Том: 0 природе
Номер: 5
Название: Земля и мы
Год выпуска: 2014
Состояние единицы хранения:
Инвентарный номер: 1235
Наличие: True

Для продолжения нажмите любую клавишу . . .

Рисунок 4.1 — Упр №4: Вывод программы

5 УПРАЖНЕНИЕ 5. РЕАЛИЗАЦИЯ МОДЕЛИ ВКЛЮЧЕНИЯ

В этом упражнении было необходимо реализовать агрегацию.

Был создан новый проект MyClassLine с классами Point и Line.

В классе Point были указаны переменные x и y , конструкторы по этим переменным и по умолчанию, вывод координат точки, метод вычисления расстояния между этой точкой и передаваемой этому методу, а также был переопределён метод ToString() для отображения точки. Этот класс можно увидеть на рисунке 5.1.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClassLine
8  {
9      Ссылка: 11
10     internal class Point
11     {
12         private double x;
13         private double y;
14
15         Ссылка: 1
16         public Point(double x, double y)
17         {
18             this.x = x;
19             this.y = y;
20         }
21
22         Ссылка: 1
23         public Point() { }
24
25         Ссылка: 2
26         public void Show()
27         {
28             Console.WriteLine("Точка с координатами: ({0}, {1})", x, y);
29         }
30
31         Ссылка: 1
32         public double Dlina(Point p)
33         {
34             double Dl = Math.Sqrt((this.x - p.x) * (this.x - p.x) +
35                                     (this.y - p.y) * (this.y - p.y));
36             return Dl;
37         }
38
39         Ссылка: 0
40         public override string ToString()
41         {
42             string ss = x + " ; " + y;
43             return ss;
44         }
45     }
46 }
```

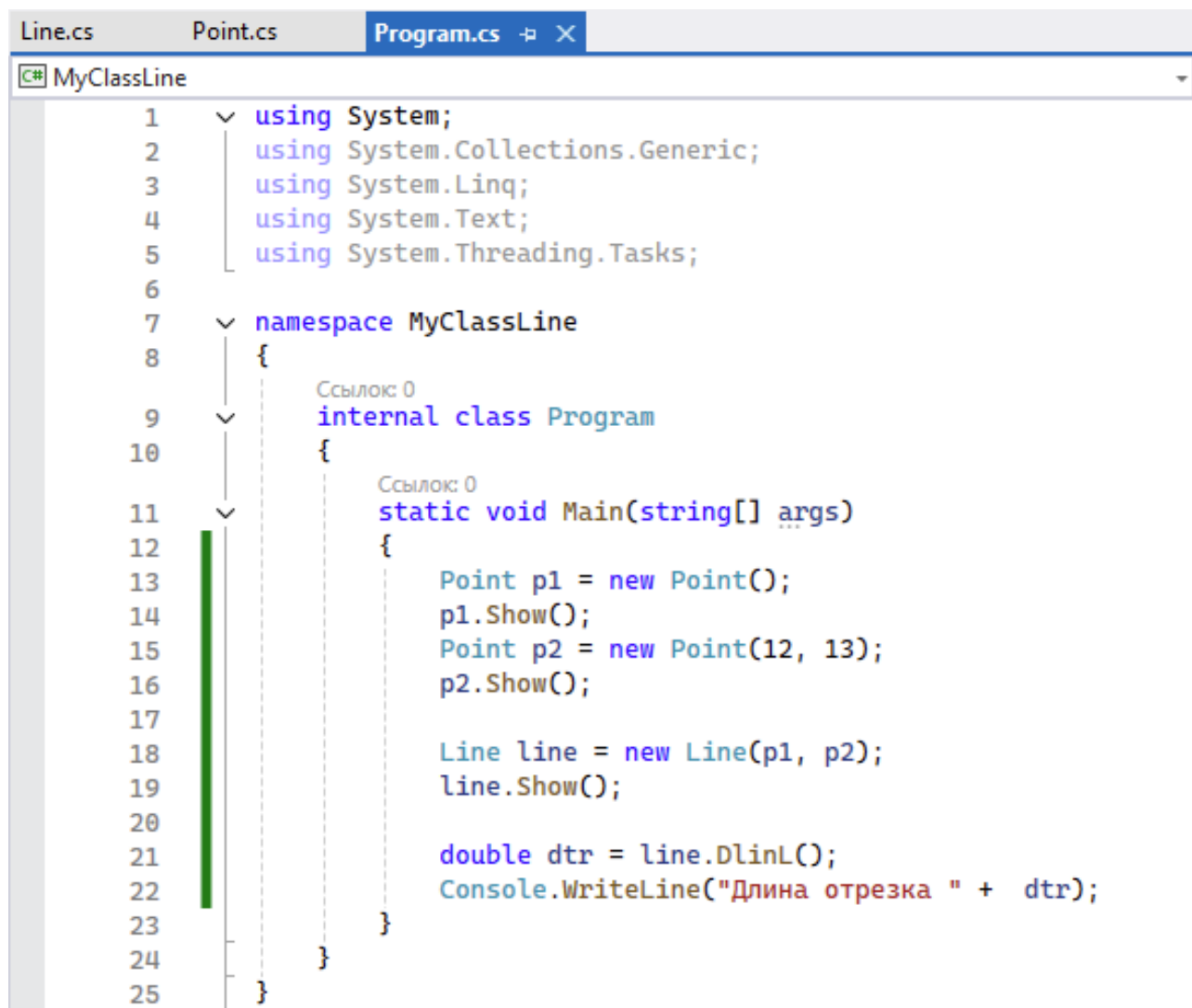
Рисунок 5.1 — Упр №5: Класс Point

В классе Line были указаны переменные начальной точки и конечной, конструкторы, вывод информации о линии и подсчёт длины отрезка, вызывающий метод нахождения расстояния между точками из класса Point. Класс Line можно увидеть на рисунке 5.2.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClassLine
8  {
9      Ссылки: 4
      internal class Line
      {
10         private Point pStart;
11         private Point pEnd;
12         Ссылки: 1
13         public Line(Point pStart, Point pEnd)
14         {
15             this.pStart = pStart;
16             this.pEnd = pEnd;
17         }
18         Ссылки: 0
19         public Line() { }
20         Ссылки: 1
21         public void Show()
22         {
23             Console.WriteLine("Отрезок с координатами: ({0}) - ({1}) ", pStart, pEnd);
24         }
25         Ссылки: 1
26         public double DlinL()
27         {
28             return pStart.Dlina(pEnd);
29         }
30     }
31 }
```

Рисунок 5.2 — Упр №5: Класс Line

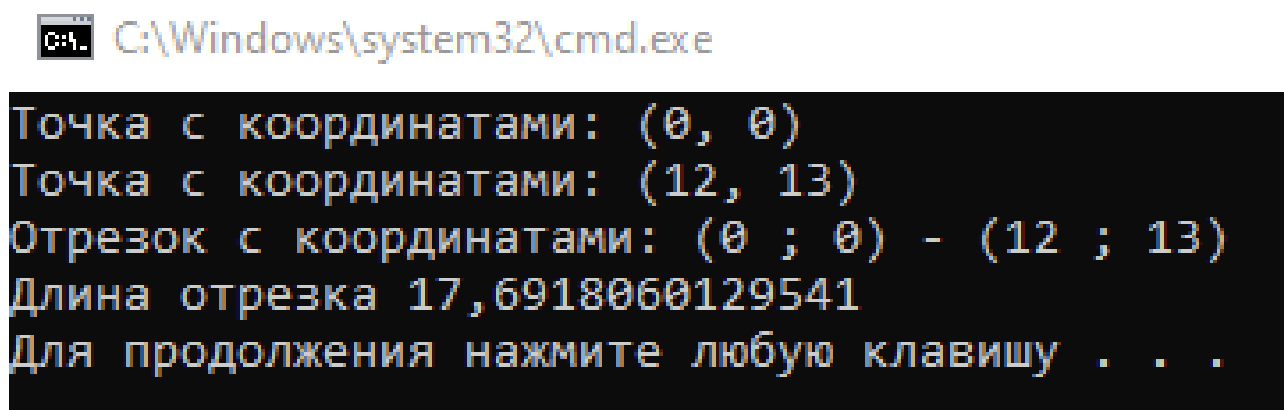
В классе Program было создано две точки и отрезок из них, была выведена вся информация о них и длина отрезка. Этот класс можно увидеть на рисунке 5.3.



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MyClassLine
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             Point p1 = new Point();
14             p1.Show();
15             Point p2 = new Point(12, 13);
16             p2.Show();
17
18             Line line = new Line(p1, p2);
19             line.Show();
20
21             double dtr = line.DlinL();
22             Console.WriteLine("Длина отрезка " + dtr);
23         }
24     }
25 }
```

Рисунок 5.3 — Упр №5: Класс Program

Результат программы можно увидеть на рисунке 5.4.



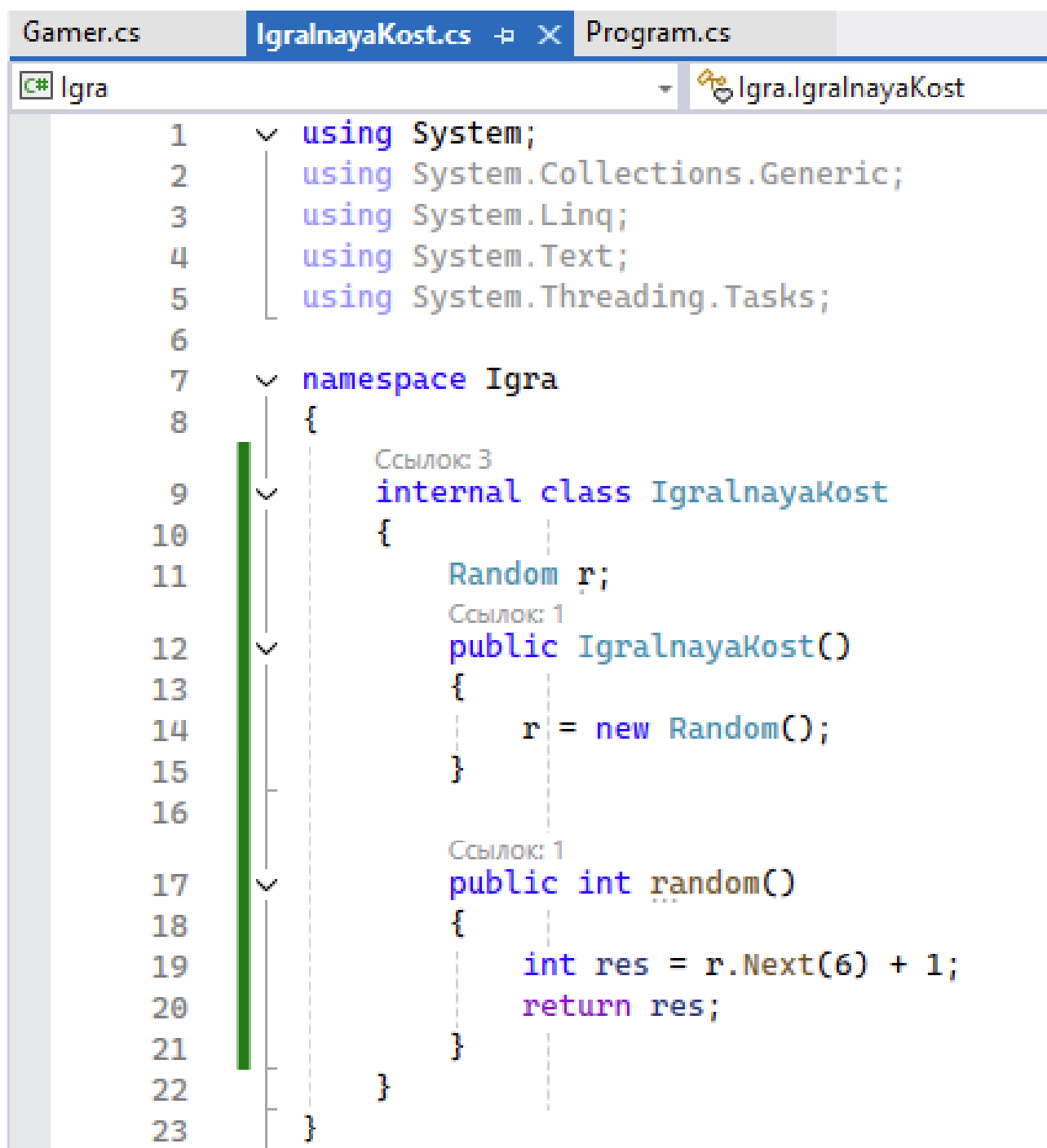
```
C:\Windows\system32\cmd.exe
Точка с координатами: (0, 0)
Точка с координатами: (12, 13)
Отрезок с координатами: (0 ; 0) - (12 ; 13)
Длина отрезка 17,6918060129541
Для продолжения нажмите любую клавишу . . .
```

Рисунок 5.4 — Упр №5: Вывод программы

6 УПРАЖНЕНИЕ 6. РЕАЛИЗАЦИЯ ОТНОШЕНИЯ АССОЦИАЦИИ МЕЖДУ КЛАССАМИ

В этом упражнении было необходимо реализовать отношение ассоциации между классами.

Был создан новый проект Igra с классами IgralnayaKost и Gamer. В первом классе был объявлен конструктор, определяющий объект-рандомизатор, и метод random(), вычисляющий случайное число от 1 до 6. Этот класс можно увидеть на рисунке 6.1.



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Igra
8  {
9      internal class IgralnayaKost
10     {
11         Random r;
12         public IgralnayaKost()
13         {
14             r = new Random();
15         }
16
17         public int random()
18         {
19             int res = r.Next(6) + 1;
20             return res;
21         }
22     }
23 }
```

Рисунок 6.1 — Упр №6: Класс IgralnayaKost

Во втором классе Gamer были объявлены поля имени и ссылки на класс игровой кости, конструктор для этих полей, где объявляется новый объект игровой кости, метод SeansGame() игры, вызывающий метод random у игровой кости, а также переопределение метода ToString(), возвращающий имя игрока. Этот класс можно увидеть на рисунке 6.2.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Igra
8  {
9      Ссылка: 3
10     internal class Gamer
11     {
12         string Name;
13         IgralnayaKost seans;
14
15         Ссылка: 1
16         public Gamer(string name)
17         {
18             Name = name;
19             seans = new IgralnayaKost();
20
21             Ссылка: 1
22             public int SeansGame()
23             {
24                 return seans.random();
25
26                 Ссылка: 1
27                 public override string ToString()
28                 {
29                     return Name;
30                 }
31             }
32         }
33     }
```

Рисунок 6.2 — Упр №6: Класс Gamer

В классе Program был назначен игрок Niko, а также создан цикл, 6 раз вызывающий метод SeansGame(). Реализация класса показана на рисунке 6.3.

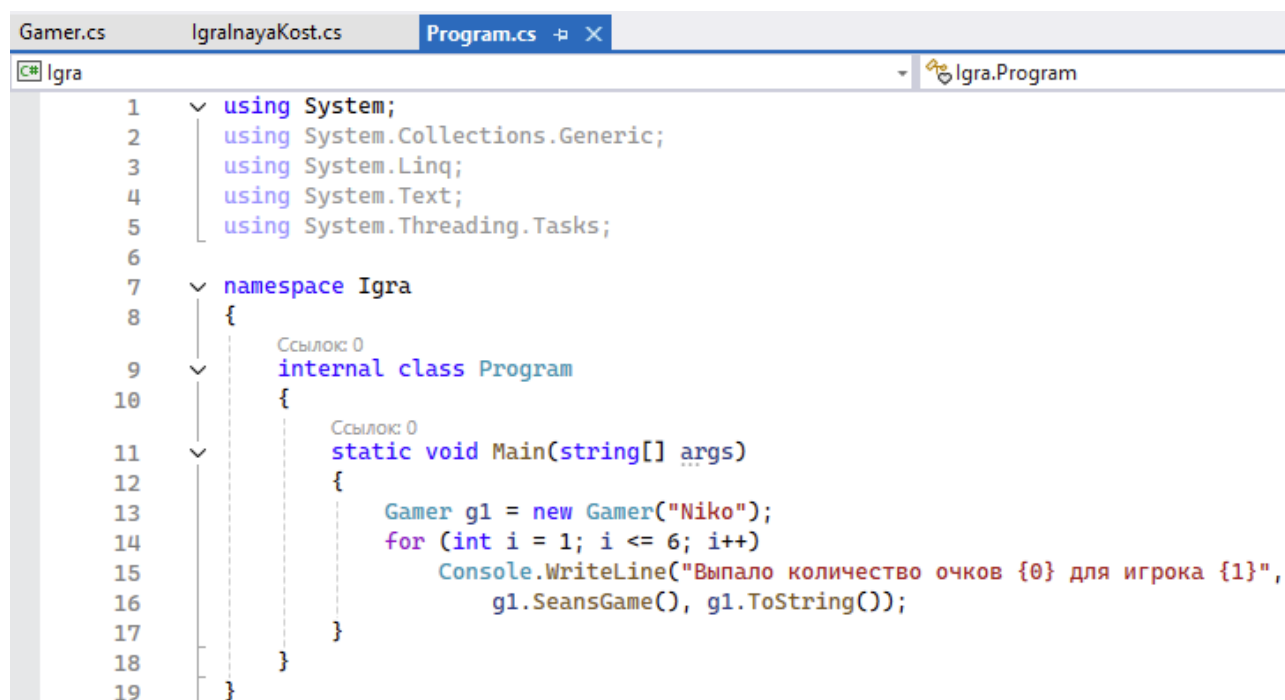


Рисунок 6.3 — Упр №6: Класс Program

Пример работы программы можно посмотреть на рисунке 6.4.

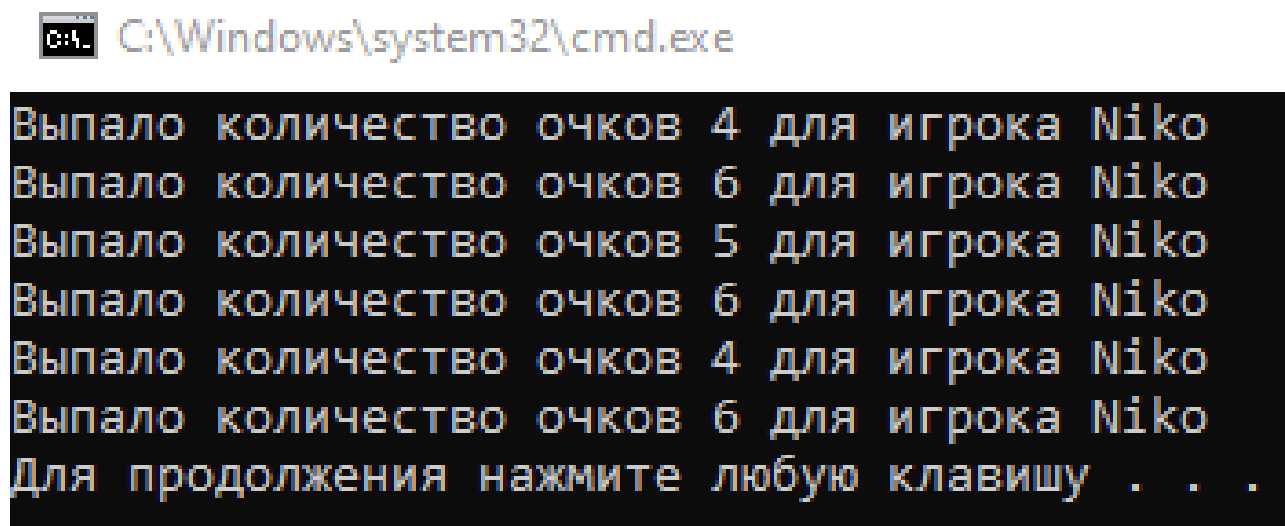


Рисунок 6.4 — Упр №6: Вывод программы

7 УПРАЖНЕНИЕ 7. РЕАЛИЗАЦИЯ ПРОГРЕССИИ

Был создан проект Progressions с абстрактным классом Progression, в котором указываются поля: diff - разница между элементами, start - первый элемент прогрессии - а также конструктор по умолчанию и для этих параметров. В нём были указаны 3 абстрактных метода: GetElement(), возвращающий элемент прогрессии с заданной позиции, PrintElement(), выводящий его и его позицию, и метод, отображающий параметры прогрессии. Этот класс показан на рисунке 7.1.

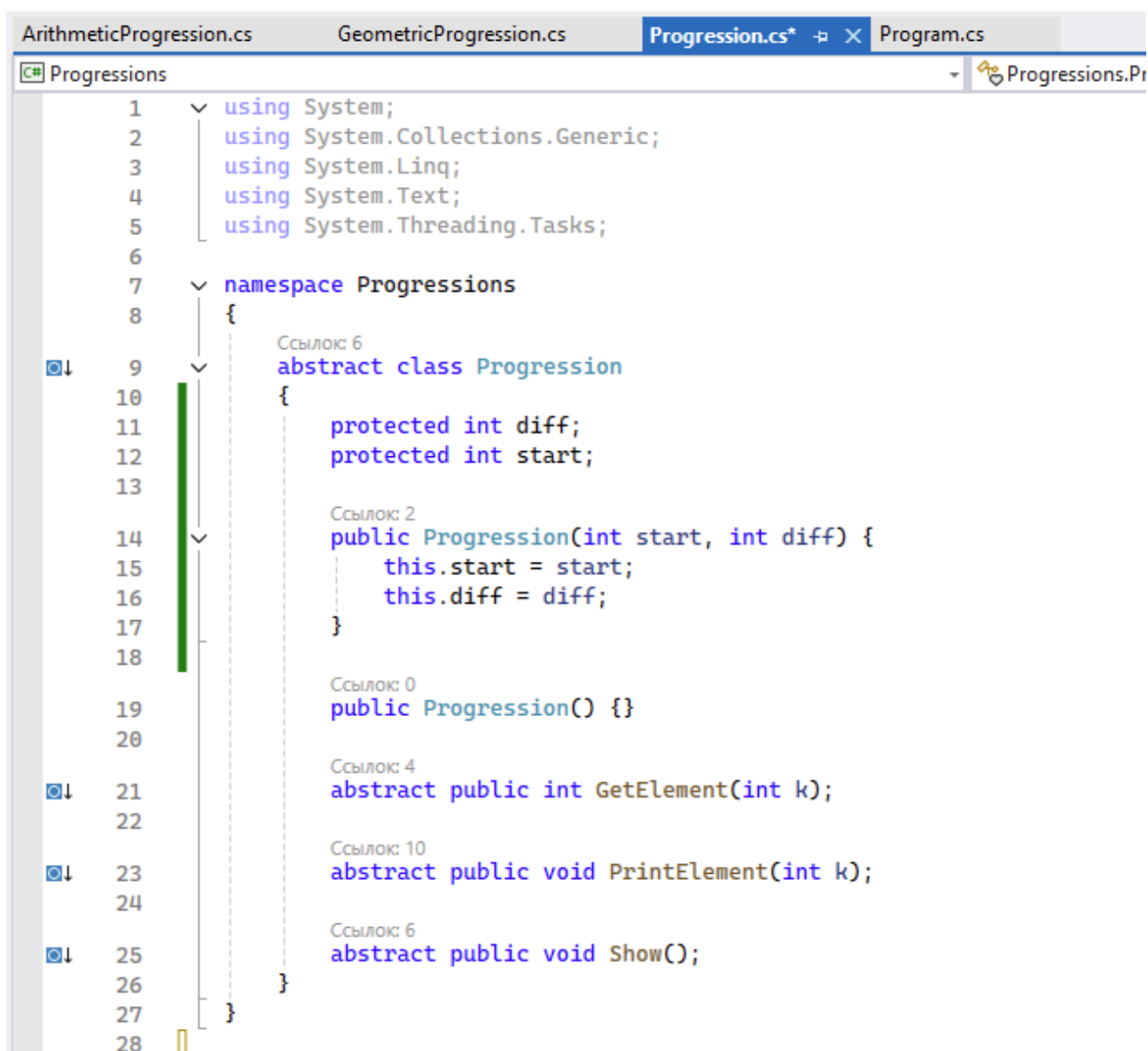


Рисунок 7.1 — Упр №7: Класс Progression

В классе ArithmeticProgression на рисунке 7.2 переобозначились все 3 метода из базового класса в соответствии с типом прогрессии. В выводах ука-

зывается, что рассматриваемая прогрессия - арифметическая, и переменные `start` и `diff` обозначаются как `a1` и `d` соответственно.

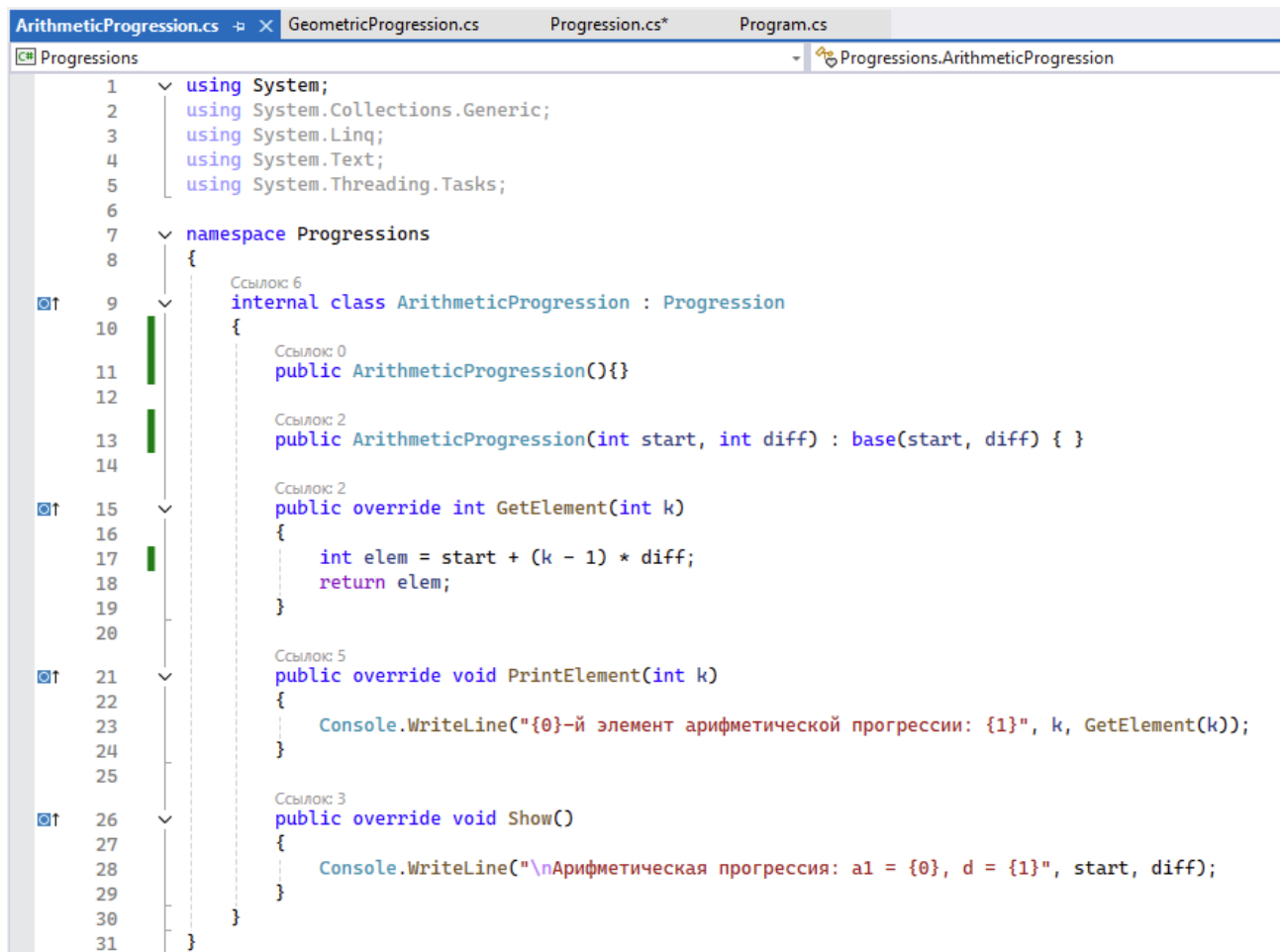


Рисунок 7.2 — Упр №7: Класс ArithmeticProgression

Переобозначение методов в классе GeometricProgression было совершено аналогичным образом, только переменные теперь обозначаются как `b1` и `q`. `GetElement()` считается через выведение `q` в нужную степень. Класс геометрической прогрессии показан на рисунке 7.3.

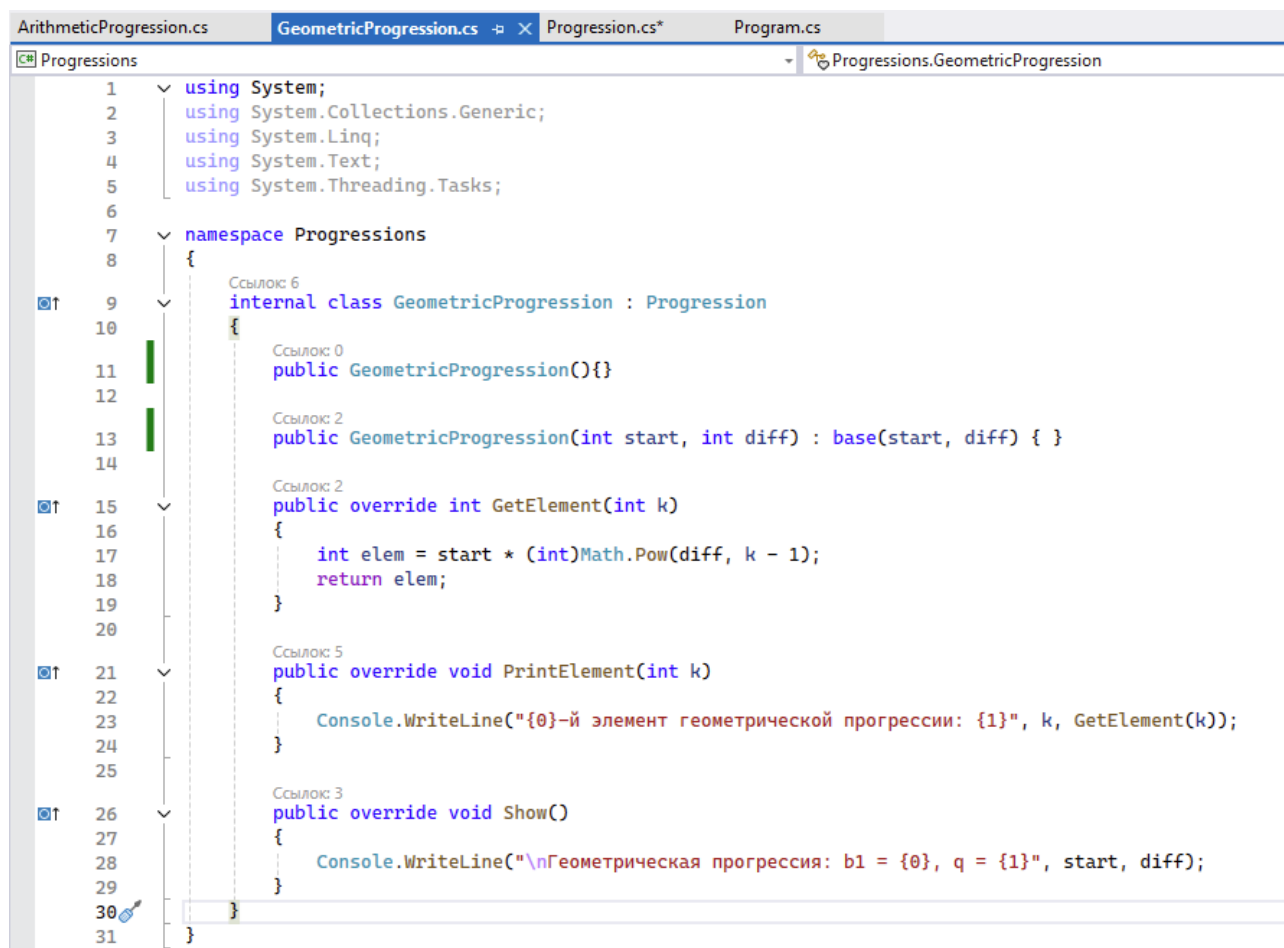


Рисунок 7.3 — Упр №7: Класс GeometricProgression

На рисунке 7.4 представлено, как создаются 4 разных прогрессии и выводятся их элементы.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Progressions
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             ArithmeticProgression a1 = new ArithmeticProgression(1, 1);
14             GeometricProgression g1 = new GeometricProgression(1, 3);
15
16             a1.Show();
17             a1.PrintElement(2);
18             a1.PrintElement(4);
19
20             g1.Show();
21             g1.PrintElement(3);
22             g1.PrintElement(5);
23
24             ArithmeticProgression a2 = new ArithmeticProgression(4, 3);
25             GeometricProgression g2 = new GeometricProgression(2, 4);
26
27             a2.Show();
28             a2.PrintElement(5);
29             a2.PrintElement(7);
30
31             g2.Show();
32             g2.PrintElement(2);
33             g2.PrintElement(4);
34         }
35     }
36 }
37
38
```

Рисунок 7.4 — Упр №7: Класс Program

С результатом программы на разных примерах можно ознакомиться на рисунке 7.4.

C:\Windows\system32\cmd.exe

Арифметическая прогрессия: $a_1 = 1$, $d = 1$

2-й элемент арифметической прогрессии: 2

4-й элемент арифметической прогрессии: 4

Геометрическая прогрессия: $b_1 = 1$, $q = 3$

3-й элемент геометрической прогрессии: 9

5-й элемент геометрической прогрессии: 81

Арифметическая прогрессия: $a_1 = 4$, $d = 3$

5-й элемент арифметической прогрессии: 16

7-й элемент арифметической прогрессии: 22

Геометрическая прогрессия: $b_1 = 2$, $q = 4$

2-й элемент геометрической прогрессии: 8

4-й элемент геометрической прогрессии: 128

Для продолжения нажмите любую клавишу . . .

Рисунок 7.5 — Упр №7: Вывод программы

ЗАКЛЮЧЕНИЕ

При выполнении лабораторной работы были созданы четыре разных проекта с различными базовыми классами и их производными, а также разными связями между ними.

Цель изучения наследования и приобретения навыков работы реализации иерархии классов была выполнена.