

**Санкт-Петербургский национальный исследовательский  
университет информационных технологий, механики и оптики**

## **ЛАБОРАТОРНАЯ РАБОТА №7**

### **Создание иерархии классов**

Студент:

*Швалов Даниил Андреевич*

*Факультет ИКТ*

*Группа: K32211*

Преподаватель:

*Иванов Сергей Евгеньевич*

## Упражнение 1. Реализация наследования классов

В проект был добавлен класс `Item`, который хранит в себе инвентарный номер `invNumber` и состояние объекта `taken`, а также имеет соответствующие методы для их изменения. Кроме того, в класс `Item` был добавлен метод, выводящий информацию на экран об единице хранения:

---

```
using System;

namespace MyClass
{
    public class Item
    {
        protected long invNumber;
        protected bool taken;

        public bool IsAvaliable()
        {
            return taken;
        }

        public long GetInvNumber()
        {
            return invNumber;
        }

        public void Take()
        {
            taken = false;
        }

        public void Return()
        {
            taken = true;
        }

        public void Show()
        {
            Console.WriteLine("Состояние единицы хранения:");
            Console.WriteLine("Инвентарный номер: {0}", invNumber);
            Console.WriteLine("Наличие: {0}", taken);
        }
    }
}
```

---

В класс `Book` добавлено наследование от класса `Item`, а также определен метод `TakeItem`:

---

```
using System;

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

        public Book() { }

        static Book()
        {
            price = 10;
        }

        public Book(
            String author,
            String title,
            String publisher,
            int pages,
            int year
        )
        {
            this.author = author;
            this.title = title;
            this.publisher = publisher;
            this.pages = pages;
            this.year = year;
        }

        public Book(String author, String title)
        {
            this.author = author;
            this.title = title;
        }

        public void SetBook(
            String author,
            String title,
            String publisher,
            int pages,
            int year
        )
        {
            this.author = author;
            this.title = title;
        }
    }
}
```

```

        this.publisher = publisher;
        this.pages = pages;
        this.year = year;
    }

    public static void SetPrice(double price)
    {
        Book.price = price;
    }

    new public void Show()
    {
        Console.WriteLine("Книга:");
        Console.WriteLine("Автор: {0}", author);
        Console.WriteLine("Название: {0}", title);
        Console.WriteLine("Год издания: {0}", year);
        Console.WriteLine("Страницы: {0}", pages);
        Console.WriteLine("Стоимость аренды: {0}", Book.price);
    }

    public double PriceBook(int s)
    {
        double cust = s * price;
        return cust;
    }

    public void TakeItem()
    {
        if (IsAvaliable())
        {
            Take();
        }
    }
}
}

```

---

В класс Program был добавлен код, проверяющий работу программы:

```

namespace MyClass;

class Program
{
    static void Main(string[] args)
    {
        Item item1 = new Item();
        item1.Show();
    }
}

```

---

На рис. 1.1 представлен пример работы программы.

Состояние единицы хранения:  
Инвентарный номер: 0  
Наличие: False

Рис. 1.1: Пример работы программы

## Упражнение 2. Использование конструкторов

В класс `Item` было добавлено два конструктора: один с аргументами, а второй по умолчанию:

---

```
using System;

namespace MyClass
{
    public class Item
    {
        protected long invNumber;
        protected bool taken;

        public Item(long invNumber, bool taken)
        {
            this.invNumber = invNumber;
            this.taken = taken;
        }

        public Item()
        {
            this.taken = true;
        }

        public bool IsAvaliable()
        {
            return taken;
        }

        public long GetInvNumber()
        {
            return invNumber;
        }

        private void Take()
        {
            taken = false;
        }
    }
}
```

```

public void Return()
{
    taken = true;
}

public void TakeItem()
{
    if (IsAvaliable())
    {
        Take();
    }
}

public void Show()
{
    Console.WriteLine("Состояние единицы хранения:");
    Console.WriteLine("Инвентарный номер: {0}", invNumber);
    Console.WriteLine("Наличие: {0}", taken);
}
}
}

```

---

В класс Book был добавлен конструктор со ссылкой на конструктор базового класса. В метод Show был добавлен вызов метода базового класса:

---

```

using System;

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

        public Book() { }

        static Book()
        {
            price = 10;
        }

        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;
    }

    public Book(String author, String title)
    {
        this.author = author;
        this.title = title;
    }

    public void SetBook(
        String author,
        String title,
        String publisher,
        int pages,
        int year
    )
    {
        this.author = author;
        this.title = title;
        this.publisher = publisher;
        this.pages = pages;
        this.year = year;
    }

    public static void SetPrice(double price)
    {
        Book.price = price;
    }

    new public void Show()
    {
        Console.WriteLine("Книга:");
        Console.WriteLine("Автор: {0}", author);
        Console.WriteLine("Название: {0}", title);
        Console.WriteLine("Год издания: {0}", year);
        Console.WriteLine("Страницы: {0}", pages);
        Console.WriteLine("Стоимость аренды: {0}", Book.price);
        base.Show();
    }

    public double PriceBook(int s)
    {

```

```

        double cust = s * price;
        return cust;
    }
}
}

```

---

В проект был добавлен класс Magazine, хранящий в себе название тома volume, его номер number, название журнала title и год выпуска year. В класс Magazine был добавлен конструктор по умолчанию, а также конструктор, принимающий данные о журнале. Кроме того, был добавлен метод Show, отображающий информацию о журнале:

---

```

using System;

namespace MyClass
{
    public class Magazine : Item
    {
        private String volume;
        private int number;
        private String title;
        private int year;

        public Magazine(
            String volume,
            int number,
            String title,
            int year,
            long invNumber,
            bool taken
        )
            : base(invNumber, taken)
        {
            this.volume = volume;
            this.number = number;
            this.title = title;
            this.year = year;
        }

        public Magazine() { }

        new public void Show()
        {
            Console.WriteLine("Журнал:");
            Console.WriteLine("Том: {0}", volume);
            Console.WriteLine("Номер: {0}", number);
            Console.WriteLine("Название: {0}", title);
            Console.WriteLine("Год выпуска: {0}", year);
            base.Show();
        }
    }
}

```



```
    }  
  }  
}
```

---

В класс Program был добавлен код, проверяющий работу программы:

---

```
namespace MyClass;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Book b = new Book(  
            "Толстой Л.Н.",  
            "Война и мир",  
            "Наука и жизнь",  
            1234,  
            2013,  
            101,  
            true  
        );  
        b.TakeItem();  
        b.Show();  
  
        Magazine m = new Magazine(  
            "О природе",  
            5,  
            "Земля и мы",  
            2014,  
            1235,  
            true  
        );  
        m.Show();  
    }  
}
```

---

На рис. 2.1 представлен пример работы программы.

Книга:  
Автор: Толстой Л.Н.  
Название: Война и мир  
Год издания: 2013  
Страницы: 1234  
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

Рис. 2.1: Пример работы программы

### Упражнение 3. Переопределение методов

В классе Item метод Return был помечен как виртуальный:

---

```
using System;

namespace MyClass
{
    public class Item
    {
        protected long invNumber;
        protected bool taken;

        public Item(long invNumber, bool taken)
        {
            this.invNumber = invNumber;
            this.taken = taken;
        }

        public Item()
        {
            this.taken = true;
        }

        public bool IsAvaliable()
        {
            return taken;
        }
    }
}
```

```

    public long GetInvNumber()
    {
        return invNumber;
    }

    private void Take()
    {
        taken = false;
    }

    public virtual void Return()
    {
        taken = true;
    }

    public void TakeItem()
    {
        if (IsAvaliable())
        {
            Take();
        }
    }

    public virtual void Show()
    {
        Console.WriteLine("Состояние единицы хранения:");
        Console.WriteLine("Инвентарный номер: {0}", invNumber);
        Console.WriteLine("Наличие: {0}", taken);
    }
}

```

---

В класс Book было добавлено новое поле returnSrok, а также метод ReturnSrok, устанавливающий, что книга сдана в срок. Кроме того, был переопределен метод Return:

---

```

using System;

namespace MyClass
{
    public class Book : Item
    {
        private String author;
        private String title;
        private String publisher;
        private int pages;
        private int year;
        private static double price = 9;
    }
}

```

```

private bool returnSrok;

public Book() { }

static Book()
{
    price = 10;
}

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;
}

public Book(String author, String title)
{
    this.author = author;
    this.title = title;
}

public void SetBook(
    String author,
    String title,
    String publisher,
    int pages,
    int year
)
{
    this.author = author;
    this.title = title;
    this.publisher = publisher;
    this.pages = pages;
    this.year = year;
}

public static void SetPrice(double price)
{
    Book.price = price;
}

```

```

public override void Show()
{
    Console.WriteLine("Книга:");
    Console.WriteLine("Автор: {0}", author);
    Console.WriteLine("Название: {0}", title);
    Console.WriteLine("Год издания: {0}", year);
    Console.WriteLine("Страницы: {0}", pages);
    Console.WriteLine("Стоимость аренды: {0}", Book.price);
    base.Show();
}

public double PriceBook(int s)
{
    double cust = s * price;
    return cust;
}

public void ReturnSrok()
{
    returnSrok = true;
}

public override void Return()
{
    taken = returnSrok;
}
}
}

```

---

В классе Magazine тоже был переопределен метод Return:

---

```

using System;

namespace MyClass
{
    public class Magazine : Item
    {
        private String volume;
        private int number;
        private String title;
        private int year;

        public Magazine(
            String volume,
            int number,
            String title,
            int year,
            long invNumber,
            bool taken

```

```

    )
    : base(invNumber, taken)
    {
        this.volume = volume;
        this.number = number;
        this.title = title;
        this.year = year;
    }

    public Magazine() { }

    public override void Show()
    {
        Console.WriteLine("Журнал:");
        Console.WriteLine("Том: {0}", volume);
        Console.WriteLine("Номер: {0}", number);
        Console.WriteLine("Название: {0}", title);
        Console.WriteLine("Год выпуска: {0}", year);
        base.Show();
    }

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

```

---

В класс Program был добавлен код, проверяющий работу программы:

---

```

namespace MyClass;

class Program
{
    static void Main(string[] args)
    {
        Book b = new Book(
            "Толстой Л.Н.",
            "Война и мир",
            "Наука и жизнь",
            1234,
            2013,
            101,
            true
        );

        Magazine m = new Magazine(
            "О природе",
            5,
            "Земля и мы",

```

```

        2014,
        1235,
        true
    );

    Console.WriteLine("Тестирование полиморфизма");

    Item it = b;
    it.TakeItem();
    it.Show();

    it = m;
    it.TakeItem();
    it.Show();
}
}

```

---

На рис. 3.1 представлен пример работы программы.

```

Тестирование полиморфизма
Книга:
Автор: Толстой Л.Н.
Название: Война и мир
Год издания: 2013
Страницы: 1234
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

```

Рис. 3.1: Пример работы программы

#### Упражнение 4. Применение абстрактного класса и абстрактных методов

Класс `Item`, а также его метод `Return` был помечен как абстрактный:

---

```

using System;

```

```

namespace MyClass
{
    public abstract class Item
    {
        protected long invNumber;
        protected bool taken;

        public Item(long invNumber, bool taken)
        {
            this.invNumber = invNumber;
            this.taken = taken;
        }

        public Item()
        {
            this.taken = true;
        }

        public bool IsAvaliable()
        {
            return taken;
        }

        public long GetInvNumber()
        {
            return invNumber;
        }

        private void Take()
        {
            taken = false;
        }

        public abstract void Return();

        public void TakeItem()
        {
            if (IsAvaliable())
            {
                Take();
            }
        }

        public virtual void Show()
        {
            Console.WriteLine("Состояние единицы хранения:");
            Console.WriteLine("Инвентарный номер: {0}", invNumber);
            Console.WriteLine("Наличие: {0}", taken);
        }
    }
}

```



Остальные классы остались без изменения:

---

```
using System;

namespace MyClass
{
    public class Book : Item
    {
        private String author;
        private String title;
        private String publisher;
        private int pages;
        private int year;
        private static double price = 9;
        private bool returnSrok;

        public Book() { }

        static Book()
        {
            price = 10;
        }

        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;
        }

        public Book(String author, String title)
        {
            this.author = author;
            this.title = title;
        }

        public void SetBook(
            String author,
            String title,
            String publisher,
```

```

        int pages,
        int year
    )
    {
        this.author = author;
        this.title = title;
        this.publisher = publisher;
        this.pages = pages;
        this.year = year;
    }

    public static void SetPrice(double price)
    {
        Book.price = price;
    }

    public override void Show()
    {
        Console.WriteLine("Книга:");
        Console.WriteLine("Автор: {0}", author);
        Console.WriteLine("Название: {0}", title);
        Console.WriteLine("Год издания: {0}", year);
        Console.WriteLine("Страницы: {0}", pages);
        Console.WriteLine("Стоимость аренды: {0}", Book.price);
        base.Show();
    }

    public double PriceBook(int s)
    {
        double cust = s * price;
        return cust;
    }

    public void ReturnSrok()
    {
        returnSrok = true;
    }

    public override void Return()
    {
        taken = returnSrok;
    }
}
}

```

---

```

using System;

namespace MyClass
{
    public class Magazine : Item

```

```

{
    private String volume;
    private int number;
    private String title;
    private int year;

    public Magazine(
        String volume,
        int number,
        String title,
        int year,
        long invNumber,
        bool taken
    )
        : base(invNumber, taken)
    {
        this.volume = volume;
        this.number = number;
        this.title = title;
        this.year = year;
    }

    public Magazine() { }

    public override void Show()
    {
        Console.WriteLine("Журнал:");
        Console.WriteLine("Том: {0}", volume);
        Console.WriteLine("Номер: {0}", number);
        Console.WriteLine("Название: {0}", title);
        Console.WriteLine("Год выпуска: {0}", year);
        base.Show();
    }

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

```

---

В класс Program был добавлен код, проверяющий работу программы:

---

```

namespace MyClass;

class Program
{
    static void Main(string[] args)
    {
        Book b = new Book(

```

```

        "Толстой Л.Н.",
        "Война и мир",
        "Наука и жизнь",
        1234,
        2013,
        101,
        true
    );

    Magazine m = new Magazine(
        "О природе",
        5,
        "Земля и мы",
        2014,
        1235,
        true
    );

    Console.WriteLine("Тестирование полиморфизма");

    Item it = b;
    it.TakeItem();
    it.Return();
    it.Show();

    it = m;
    it.TakeItem();
    it.Return();
    it.Show();
}
}

```

---

На рис. 4.1 представлен пример работы программы.

Тестирование полиморфизма  
Книга:  
Автор: Толстой Л.Н.  
Название: Война и мир  
Год издания: 2013  
Страницы: 1234  
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

Рис. 4.1: Пример работы программы

## Упражнение 5. Реализации модели включения

В проект был добавлен класс `Point`, содержащий координаты  $x$  и  $y$ . Также в класс был добавлен конструктор, принимающий эти координаты. Кроме того, в класс были добавлены методы `Show` для отображения информации о точке, `Dlina` для вычисления расстояния между двумя точками и `ToString` для отображения объекта:

---

```
using System;

namespace MyClassLine
{
    public class Point
    {
        private double x;
        private double y;

        public Point(double x, double y)
        {
            this.x = x;
            this.y = y;
        }

        public Point() { }

        public void Show()
```

```

    {
        Console.WriteLine("Точка с координатами: ({0}, {1})", x, y);
    }

    public double Dlina(Point p)
    {
        return Math.Sqrt((x - p.x) * (x - p.x) + (y - p.y) * (y - p.y));
    }

    public override string ToString()
    {
        return x + " ; " + y;
    }
}

```

---

В проект был добавлен класс Line, содержащий начальную pStart и конечную pEnd точки. Также в класс был добавлен конструктор, принимающий начальную и конечную точку. Кроме того, были добавлены методы Show для отображения информации об отрезке, а также DlinL для вычисления длины отрезка:

---

```

using System;

namespace MyClassLine
{
    public class Line
    {
        private Point pStart;
        private Point pEnd;

        public Line(Point pStart, Point pEnd)
        {
            this.pStart = pStart;
            this.pEnd = pEnd;
        }

        public Line() { }

        public void Show()
        {
            Console.WriteLine(
                "Отрезок с координатами: ({0}) - ({1})",
                pStart,
                pEnd
            );
        }

        public double DlinL()
        {

```

```

        return pStart.Dlina(pEnd);
    }
}

```

---

В класс Program был добавлен код, проверяющий работу программы:

```

namespace MyClassLine;

class Program
{
    static void Main(string[] args)
    {
        Point p1 = new Point();
        p1.Show();
        Point p2 = new Point(12, 13);
        p2.Show();

        Line line = new Line(p1, p2);
        line.Show();

        double dtr = line.DlinL();
        Console.WriteLine("Длина отрезка {0}", dtr);
    }
}

```

---

На рис. 5.1 представлен пример работы программы.

```

Точка с координатами: ({0}, {1})
Точка с координатами: ({0}, {1})
Отрезок с координатами: (0 ; 0) - (12 ; 13)
Длина отрезка 17.69180601295413

```

Рис. 5.1: Пример работы программы

## Упражнение 6. Реализация отношения ассоциации между классами

В проект был добавлен класс IgralnayaKost, хранящий генератор псевдослучайных чисел r. Для него был создан конструктор по умолчанию, создающий генератор псевдослучайных чисел. Также в класс был добавлен метод random, возвращающий случайное число в диапазоне от 1 до 6:

```

using System;

namespace Igra
{
    public class IgralnayaKost

```

```

{
    Random r;

    public IgralnayaKost()
    {
        r = new Random();
    }

    public int Random()
    {
        return r.Next(6) + 1;
    }
}

```

---

В проект был добавлен класс Gamer, хранящий имя name и игральную кость seans. Для него был реализован конструктор, принимающий имя игрока. Кроме того, были добавлены методы SeansGame для броска кости и ToString для отображения объекта:

---

```

using System;

namespace Igra
{
    public class Gamer
    {
        private string name;
        private IgralnayaKost seans;

        public Gamer(string name)
        {
            this.name = name;
            seans = new IgralnayaKost();
        }

        public int SeansGame()
        {
            return seans.Random();
        }

        public override string ToString()
        {
            return name;
        }
    }
}

```

---

В класс Program был добавлен код, проверяющий работу программы:



---

```
namespace Igra;

class Program
{
    static void Main(string[] args)
    {
        Gamer g1 = new Gamer("Niko");

        for (int i = 1; i ≤ 6; ++i)
        {
            Console.WriteLine(
                "Выпало количество очков {0} для игрока {1}",
                g1.SeansGame(),
                g1.ToString()
            );
        }
    }
}
```

---

На рис. 6.1 представлен пример работы программы.

```
Выпало количество очков 4 для игрока Niko
Выпало количество очков 4 для игрока Niko
Выпало количество очков 5 для игрока Niko
Выпало количество очков 1 для игрока Niko
Выпало количество очков 4 для игрока Niko
Выпало количество очков 2 для игрока Niko
```

Рис. 6.1: Пример работы программы

## Упражнение 7. Реализация прогрессии

В проект был добавлен абстрактный класс `Progression`, в который был добавлен абстрактный метод `GetElement`, возвращающий  $k$ -ый элемент прогрессии:

---

```
using System;

namespace Progression
{
    public abstract class Progression
    {
        public abstract double GetElement(int k);
    }
}
```

---

В проект был добавлен класс `ArithmeticProgression`, реализующий арифметическую прогрессию. Конструктор этого класса принимает первый член прогрессии `firstMember` и разность прогрессии `difference`:

---

```
using System;

namespace Progression
{
    public class ArithmeticProgression : Progression
    {
        private double firstMember;
        private double difference;

        public ArithmeticProgression(double firstMember, double difference)
        {
            this.firstMember = firstMember;
            this.difference = difference;
        }

        public override double GetElement(int k)
        {
            return firstMember + (k - 1) * difference;
        }
    }
}
```

---

В проект также был добавлен класс `GeometricProgression`, представляющий геометрическую прогрессию. Конструктор этого класса принимает первый член прогрессии `firstMember` и знаменатель прогрессии `denominator`:

---

```
using System;

namespace Progression
{
    public class GeometricProgression : Progression
    {
        private double firstMember;
        private double denominator;

        public GeometricProgression(double firstMember, double denominator)
        {
            this.firstMember = firstMember;
            this.denominator = denominator;
        }

        public override double GetElement(int k)
        {
            return firstMember * Math.Pow(denominator, k - 1);
        }
    }
}
```

```
    }  
  }  
}
```

---

В класс Program был добавлен код, проверяющий работу программы:

---

```
namespace Progression;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        ArithmeticProgression a = new ArithmeticProgression(1, 5);  
        Console.WriteLine("Arithmetic progression: {0}", a.GetElement(5));  
  
        GeometricProgression g = new GeometricProgression(1, 5);  
        Console.WriteLine("Geometric progression: {0}", g.GetElement(5));  
    }  
}
```

---

На рис. 7.1 представлен пример работы программы.

Arithmetic progression: 21  
Geometric progression: 625

Рис. 7.1: Пример работы программы