

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

ЛАБОРАТОРНАЯ РАБОТА №8

**Использование интерфейсов при
реализации иерархии классов**

Студент:

Швалов Даниил Андреевич

Факультет ИКТ

Группа: К32211

Преподаватель:

Иванов Сергей Евгеньевич

Упражнение 1. Создание и реализация интерфейса

В проект был добавлен интерфейс IPubs, содержащий метод Subs для проверки подписки и свойство IfSubs для оформления подписки:

```
using System;

namespace MyClass
{
    public interface IPubs
    {
        void Subs();
        bool IfSubs { get; set; }
    }
}
```

Класс Magazine был унаследован от интерфейса IPubs. Также были реализованы свойства и методы, объявленные в интерфейсе:

```
using System;

namespace MyClass
{
    public class Magazine : Item, IPubs
    {
        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;
}

public bool IfSubs { get; set; }

public void Subs()
{
    Console.WriteLine(
        "Подписка на журнал \"{0}\": {1}.",
        title,
        IfSubs
    );
}
}
}

```

Классы Book и Item остались без изменений:

```

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 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);
    }
}
}

```

В класс Program был добавлен код, проверяющий работу программы:

```

namespace MyClass;

class Program
{
    static void Main(string[] args)
    {
        Magazine m = new Magazine(
            "О природе",
            5,
            "Земля и мы",
            2014,
            1235,
            true
        );
        m.TakeItem();
        m.Show();
        m.IfSubs = true;
        m.Subs();
    }
}

```

На рис. 1.1 представлен пример работы программы.

```
Журнал:
Том: 0 природе
Номер: 5
Название: Земля и мы
Год выпуска: 2014
Состояние единицы хранения:
Инвентарный номер: 1235
Наличие: False
Подписка на журнал "Земля и мы": True.
```

Рис. 1.1: Пример работы программы

Упражнение 2. Использование стандартных интерфейсов

В класс `Item` было добавлено наследование интерфейса `IComparable`, а также был реализован метод этого интерфейса `CompareTo` по полю `invNumber`:

```
using System;

namespace MyClass
{
    public abstract class Item : IComparable
    {
        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);
}

int IComparable.CompareTo(object obj)
{
    Item it = (Item)obj;
    if (invNumber == it.invNumber)
    {
        return 0;
    }
    else if (invNumber > it.invNumber)
    {
        return 1;
    }
    else
    {
        return -1;
    }
}
}
}

```

Остальные классы остались без изменения:

```

using System;

namespace MyClass
{
    public interface IPubs
    {
        void Subs();
    }
}

```



```
        bool IfSubs { get; set; }  
    }  
}
```

```
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, IPubs
    {
        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;
        }

        public bool IfSubs { get; set; }

        public void Subs()
        {
            Console.WriteLine(
                "Подписка на журнал \"{0}\": {1}.",
                title,
                IfSubs
            );
        }
    }
}

```

```
}  
}
```

В класс Program был добавлен код, проверяющий работу программы:

```
namespace MyClass;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Item[] items = new Item[4];  
  
        Book b = new Book();  
        b.SetBook("Пушкин А.С.", "Капитанская дочка", "Вильямс", 123, 2012);  
        items[0] = b;  
        items[1] = new Book(  
            "Толстой Л.Н.",  
            "Война и мир",  
            "Наука и жизнь",  
            1234,  
            2013,  
            101,  
            true  
        );  
        items[2] = new Book("Лермонтов М.Ю.", "Мцыри");  
        items[3] = new Magazine("О природе", 5, "Земля и мы", 2014, 1235, true);  
  
        Array.Sort(items);  
  
        Console.WriteLine("Сортировка по инвентарному номеру:");  
        foreach (Item item in items)  
        {  
            item.Show();  
        }  
    }  
}
```

На рис. 2.1 представлен пример работы программы.

Сортировка по инвентарному номеру:
Книга:
Автор: Пушкин А.С.
Название: Капитанская дочка
Год издания: 2012
Страницы: 123
Стоимость аренды: 10
Состояние единицы хранения:
Инвентарный номер: 0
Наличие: True
Книга:
Автор: Лермонтов М.Ю.
Название: Мцыри
Год издания: 0
Страницы: 0
Стоимость аренды: 10
Состояние единицы хранения:
Инвентарный номер: 0
Наличие: True
Книга:
Автор: Толстой Л.Н.
Название: Война и мир
Год издания: 2013
Страницы: 1234
Стоимость аренды: 10
Состояние единицы хранения:
Инвентарный номер: 101
Наличие: True
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. Реализация прогрессии с помощью интерфейса

Абстрактный класс `Progression` был заменен на интерфейс `IProgression`:

```
using System;
```

```
namespace Progression
{
    public interface IProgression
    {
        double GetElement(int k);
    }
}
```

Классы ArithmeticProgression и GeometricProgression были переделаны под использование этого интерфейса:

```
using System;
```

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

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

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

```
using System;
```

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

        public GeometricProgression(double firstMember, double difference)
        {
            this.firstMember = firstMember;
```

```
        this.denominator = difference;
    }

    double IProgression.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)
    {
        IProgression a = new ArithmeticProgression(1, 5);
        Console.WriteLine("Arithmetic progression: {0}", a.GetElement(5));

        IProgression g = new GeometricProgression(1, 5);
        Console.WriteLine("Geometric progression: {0}", g.GetElement(5));
    }
}
```

На рис. 3.1 представлен пример работы программы.

Arithmetic progression: 21
Geometric progression: 625

Рис. 3.1: Пример работы программы