

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

Кафедра программных систем

**Лабораторная работа №7**

**Создание иерархии классов**

Выполнил: Мороз

Илья Олегович

Группа № K3120

Проверила: Казанова П.П.

Санкт-Петербург

2019

**Цель работы:**

Целью работы является знакомство с отношениями между классами в языке C# и их применение на практике.

**Ход работы:**

**Упражнение 1.**

1. За основу был взят проект из предыдущей лабораторной работы MyClass.
2. Был создан класс-родитель Item.
3. Для класса Item были описаны поля и методы.
4. В классе Book было указано наследование от класса Item.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Item

{

protected long invNumber;

protected bool taken;

public bool IsAvaible()

{

if (taken)

return true;

else

return false;

}

public long GetInvNumber()

{

return invNumber;

}

public void Take()

{

taken = false;

}

public void Return()

{

taken = true;

}

public void Print()

{

Console.WriteLine("Состояние единицы хранения:\n Инвентарный номер: {0}\n Наличие: {1}", invNumber, taken);

}

}

}

Листинг 1 – класс Item

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Book : Item

{

public string Author { get; set; }

public string Title { get; set; }

public string Publisher { get; set; }

public int Pages { get; set; }

public int Year { get; set; }

public static double Price { get; set; }

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 string ToString()

{

string bs = String.Format("\nКнига:\n Автор: {0}\n Название: {1}\n Год издания: {2}\n {3} стр.\n Стоимость аренды: {4}", Author, Title, Year, Pages, Price);

return bs;

}

new public void Print()

{

Console.WriteLine(this);

}

public double PriceBook(int s)

{

double cust = s \* Price;

return cust;

}

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() { }

static Book()

{

Price = 9;

}

public Book(string author, string title)

{

this.Author = author;

this.Title = title;

}

public void TakeItem()

{

if (this.IsAvaible())

{

this.Take();

}

}

}

}

Листинг 2 – класс Book

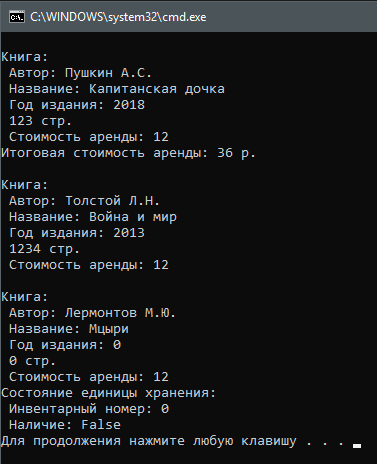


Рисунок 1 – результат работы программы

**Упражнение 2.**

1. Для класса Item были написаны два конструктора.
2. В классе Book были внесены изменения в конструктор. Теперь происходит вызов конструктора родителя при создании объекта класса Book.
3. В метод Print был добавлен вызов метода родителя.

public Item(long invNumber, bool taken)

{

this.invNumber = invNumber;

this.taken = taken;

}

public Item()

{

this.taken = true;

}

Листинг 3 – конструкторы класса Item

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;

}

Листинг 4 – изменённый конструктор класса Book

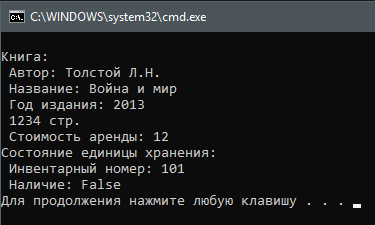


Рисунок 2 – результат работы программы

1. Был создан новый дочерний класс Magazine.
2. Для данного класса были описаны его характеристики – поля и методы.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Magazine : Item

{

public string Volume { get; set; }

public int Number { get; set; }

public string Title { get; set; }

public int Year { get; set; }

public Magazine(string volume, int number, string title, int year, long invNumber, bool taken) : base(invNumber, taken)

{

Volume = volume;

Number = number;

Title = title;

Year = year;

}

public Magazine() { }

public override string ToString()

{

string bs = String.Format("\nЖурнал:\n Том: {0}\n Номер: {1}\n Название: {2}\n Год выпуска: {3}", Volume, Number, Title, Year);

return bs;

}

new public void Print()

{

Console.WriteLine(this);

base.Print();

}

}

}

Листинг 5 – класс Magazine

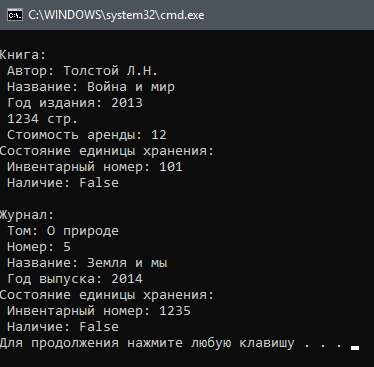


Рисунок 3 – результат работы программы

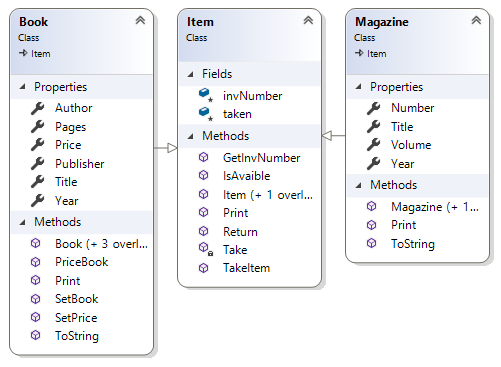


Рисунок 4 – полученная диаграмма классов

**Упражнение 3.**

1. В данном упражнении был рассмотрен полиморфизм на основе абстрактных классов.
2. В родительском классе методы Return и Print были объявлены как абстрактные.
3. В дочерних классах данные методы были переопределены с ключевым словом override.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Item

{

protected long invNumber;

protected bool taken;

public bool IsAvaible()

{

if (taken)

return true;

else

return false;

}

public long GetInvNumber()

{

return invNumber;

}

private void Take()

{

taken = false;

}

public virtual void Return()

{

taken = true;

}

public virtual void Print()

{

Console.WriteLine("Состояние единицы хранения:\n Инвентарный номер: {0}\n Наличие: {1}", invNumber, taken);

}

public Item(long invNumber, bool taken)

{

this.invNumber = invNumber;

this.taken = taken;

}

public Item()

{

this.taken = true;

}

public void TakeItem()

{

if (this.IsAvaible())

{

this.Take();

}

}

}

}

Листинг 6 – класс Item

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Book : Item

{

public string Author { get; set; }

public string Title { get; set; }

public string Publisher { get; set; }

public int Pages { get; set; }

public int Year { get; set; }

public static double Price { get; set; }

public bool returnSrok { get; private set; }

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 string ToString()

{

string bs = String.Format("\nКнига:\n Автор: {0}\n Название: {1}\n Год издания: {2}\n {3} стр.\n Стоимость аренды: {4}", Author, Title, Year, Pages, Price);

return bs;

}

public override void Print()

{

Console.WriteLine(this);

base.Print();

}

public double PriceBook(int s)

{

double cust = s \* Price;

return cust;

}

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() { }

static Book()

{

Price = 9;

}

public Book(string author, string title)

{

this.Author = author;

this.Title = title;

}

public void ReturnSrok()

{

returnSrok = true;

}

public override void Return()

{

if (returnSrok == true)

{

taken = true;

}

else

{

taken = false;

}

}

}

}

Листинг 7 – класс Book

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Magazine : Item

{

public string Volume { get; set; }

public int Number { get; set; }

public string Title { get; set; }

public int Year { get; set; }

public Magazine(string volume, int number, string title, int year, long invNumber, bool taken) : base(invNumber, taken)

{

Volume = volume;

Number = number;

Title = title;

Year = year;

}

public Magazine() { }

public override string ToString()

{

string bs = String.Format("\nЖурнал:\n Том: {0}\n Номер: {1}\n Название: {2}\n Год выпуска: {3}", Volume, Number, Title, Year);

return bs;

}

public override void Print()

{

Console.WriteLine(this);

base.Print();

}

public override void Return()

{

taken = true;

}

}

}

Листинг 8 – класс Magazine

Console.WriteLine("\nТестирование полиморфизма:");

Item it;

it = b2;

it.TakeItem();

it.Print();

it = mag1;

it.TakeItem();

it.Print();

Листинг 9 – код для тестирования полиморфизма

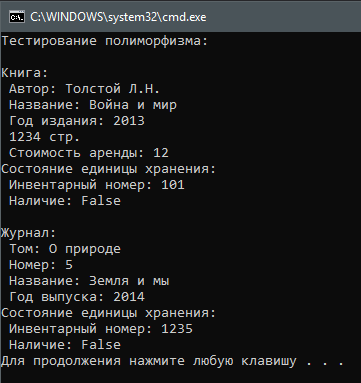


Рисунок 5 – тестирование полиморфизма

**Упражнение 4.**

1. Класс Item был объявлен как абстрактный. Также его метод Return стал абстрактным.
2. В методе Main была проверена работоспособность кода после внесённых изменений.

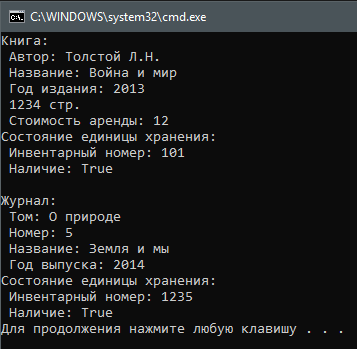


Рисунок 6 – код работает

**Упражнение 5.**

1. Для реализации связи Ассоциация между классами был создан новый класс Publisher.
2. Для класса Publisher были описаны поля и методы.
3. Класс Publisher был связан с классом Book с помощью связи ассоциация.
4. В методе Main их связь была протестирована.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Publisher

{

struct LicenseNumber

{

public int licenseNumber;

public DateTime data;

public override string ToString()

{

string bs = String.Format("Лицензия N-{0} от {1} г.", licenseNumber, data.Year);

return bs;

}

}

public string Name { get; set; }

public string EmailAdress { get; set; }

private LicenseNumber LinNumber;

public Publisher(string name, string emailAdress, int lnumber, DateTime data)

{

this.Name = name;

this.EmailAdress = emailAdress;

this.LinNumber.licenseNumber = lnumber;

this.LinNumber.data = data;

}

public override string ToString()

{

string bs = String.Format(": {0}, электронный адрес: {1}, {2}", this.Name, this.EmailAdress, this.LinNumber);

return bs;

}

}

}

Листинг 10 – класс Publisher

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Book : Item

{

public string Author { get; set; }

public string Title { get; set; }

//public string Publisher { get; set; }

public Publisher Publ { get; set; }

public int Pages { get; set; }

public int Year { get; set; }

public static double Price { get; set; }

public bool returnSrok { get; private set; }

public void SetBook(string author, string title, Publisher publisher, int pages, int year)

{

this.Author = author;

this.Title = title;

this.Publ = publisher;

this.Pages = pages;

this.Year = year;

}

public static void SetPrice(double price)

{

Book.Price = price;

}

public override string ToString()

{

string bs = String.Format("\nКнига:\n Автор: {0}\n Название: {1}\n Год издания: {2}\n {3} стр.\n Стоимость аренды: {4}\n Издательство{5}", Author, Title, Year, Pages, Price, Publ.ToString());

return bs;

}

public override void Print()

{

Console.WriteLine(this);

base.Print();

}

public double PriceBook(int s)

{

double cust = s \* Price;

return cust;

}

public Book(string author, string title, Publisher publisher, int pages, int year, long invNumber, bool taken) : base(invNumber, taken)

{

this.Author = author;

this.Title = title;

this.Publ = publisher;

this.Pages = pages;

this.Year = year;

}

public Book() { }

static Book()

{

Price = 9;

}

public Book(string author, string title)

{

this.Author = author;

this.Title = title;

}

public void ReturnSrok()

{

returnSrok = true;

}

public override void Return()

{

if (returnSrok == true)

{

taken = true;

}

else

{

taken = false;

}

}

}

}

Листинг 11 – изменённый класс Book

Book.SetPrice(12);

Publisher publ = new Publisher("Наука и жизнь", "nauka@mail.ru", 1234, new DateTime(2014, 12, 14));

Book b2 = new Book("Толстой Л.Н.", "Война и мир", publ, 1234, 2013, 101, true);

b2.TakeItem();

b2.Print();

Листинг 12 – код для проверки связи между классами

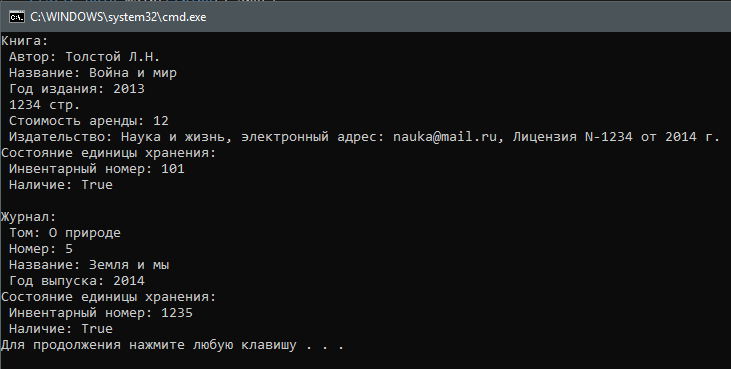


Рисунок 7 – проверка работоспособности программы

**Упражнение 6.**

1. Был создан новый проект Progression для работы с арифметическими и геометрическими прогрессиями.
2. Был создан абстрактный класс Progression с абстрактным методом GetElement.
3. Были созданы два класса-наследника ArithmeticProgression и GeometricProgression. В этих классах были реализованы конструкторы, а также был перегружен метод GetElement.
4. В методе Main были созданы объекты данных классов, на которых была произведена отладка программы.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

abstract class Progression

{

public abstract double GetElement(int k);

}

}

Листинг 13 – класс Progression

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

class ArithmeticProgression : Progression

{

public int a1 { get; private set; }

public int d { get; private set; }

public ArithmeticProgression(int a1, int d)

{

this.a1 = a1;

this.d = d;

}

public override double GetElement(int k)

{

double element = this.a1 + this.d \* (k - 1);

return element;

}

}

}

Листинг 14 – класс ArithmeticProgression

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

class GeometricProgression : Progression

{

public int b1 { get; private set; }

public int q { get; private set; }

public GeometricProgression(int b1, int q)

{

this.b1 = b1;

this.q = q;

}

public override double GetElement(int k)

{

double element = this.b1 \* Math.Pow(this.q, k - 1);

return element;

}

}

}

Листинг 15 – класс GeometricProgression

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

class Program

{

static void Main(string[] args)

{

ArithmeticProgression arProg = new ArithmeticProgression(3, 12);

GeometricProgression geoProg = new GeometricProgression(2, 3);

Console.Write("Enter number of element: ");

int n = int.Parse(Console.ReadLine());

Console.WriteLine("Element in arithmetic progression is {0}\nElement in geometric progression is {1}", arProg.GetElement(n), geoProg.GetElement(n));

}

}

}

Листинг 16 – класс Program

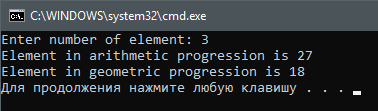


Рисунок 8 – результат работы программы

**Вывод:**

В ходе выполнения данной лабораторной работы были получены новые знания в языке C#. Были рассмотрены отношения между классами, абстрактные методы и классы, перегрузка абстрактных методов. Полученные знания были закреплены написанием программы для работы с прогрессиями.

**Ответы на контрольные вопросы:**

1. Для чего можно создавать конструктор у абстрактного класса?

Для того, чтобы при наследовании от абстрактного класса вызывать конструктор родителя, например public Magazine(string volume, int number, string title, int year, long invNumber, bool taken) : base(invNumber, taken).

1. Как хранятся переменные, полученные в методах?

Память для данных переменных выделяется при вызове метода, а после его работы удаляется.