Университет ИТМО

**Лабораторная работа #8**

**Использование интерфейсов при реализации иерархии классов**

Выполнил: Михайлов

Иван Юрьевич

Группа № К3121

Проверила: Филимонова И. А.

Санкт-Петербург

2020

**Цель работы:**

Рассмотреть способы работы с интерфейсами в языке C#.

**Ход работы:**

**Упражнение 1.** Создание и реализация интерфейса.

Для реализации разной функциональности взятия оплаты за пользование книгами и журналами, для этих классов созданы собственные интерфейсы, определяющие поведения этих классов.

Для класса Book создан интерфейс IPr, в котором были объявлены его функциональные члены – методы установки базовой стоимости и расчета общей стоимости.

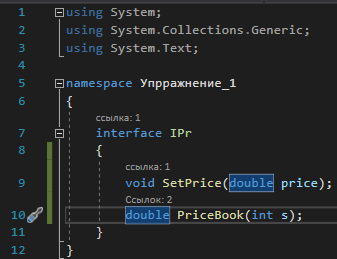


Рисунок 1 – Интерфейс IPr

В класс Book добавлен интерфейс IPr в список наследования и реализованы методы, объявленные в интерфейсе.

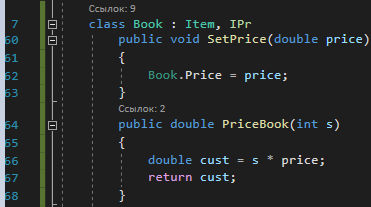


Рисунок 2 – Изменения в классе Book

Для класса Magazin создан еще один интерфейс IPubs, который реализует метод Sub, для оформления подписка, и свойство IfSubs, для проверки факта оформления подписки на издание

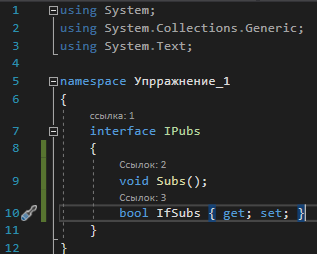


Рисунок 3 – Интерфейс IPubs

В класс Magazine добавлен интерфейс IPubs в список наследования и реализованы методы, объявленные в интерфейсе.

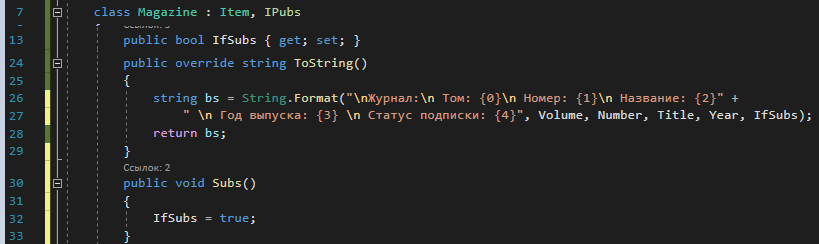


Рисунок 4 – Изменения в классе Magazine

В методе Main класса Program для уже имеющегося журнала mag1 вызван метод Subs для отображения информации о подписке.

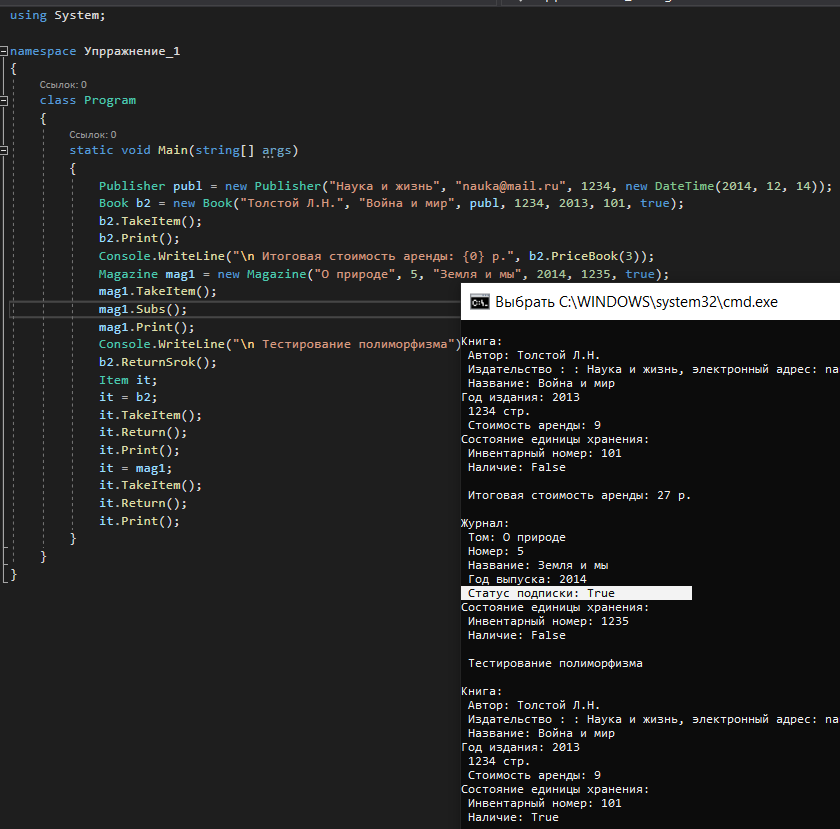


Рисунок 5 – Результат работы программы

**Упражнение 2.** Использование стандартных интерфейсов.

В объявление абстрактного класса Item добавлено наследование стандартного интерфейса IComparable, определённого в библиотеке классов .Net, и содержащий единственный метод CompareTo, возвращающий результат сравнения двух объектов – текущего и переданного ему в качестве параметра. Реализация данного метода должна возвращать: 0 – если текущий объект и параметр равны, отрицательное число, если текущий объект меньше параметра и положительное число, если текущий объект больше параметра. В класс Item добавлена реализация этого метода, причем сравнение происходит по полю invNumber.

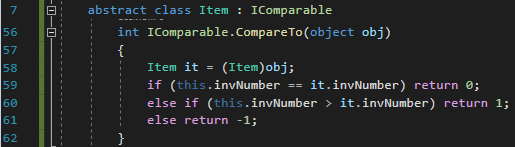


Рисунок 6 – Изменения в классе Item

В методе Main класса Program добавлена директива using для пространства имен System.Collection.Generic. Создан список ссылок на абстрактный базовый класс Item. В него были добавлены и отсортированы 3 книги и 2 журнала с помощью метода AddRange. Использован полиморфный вызов метода Print для отображения всего списка книг и журналов.

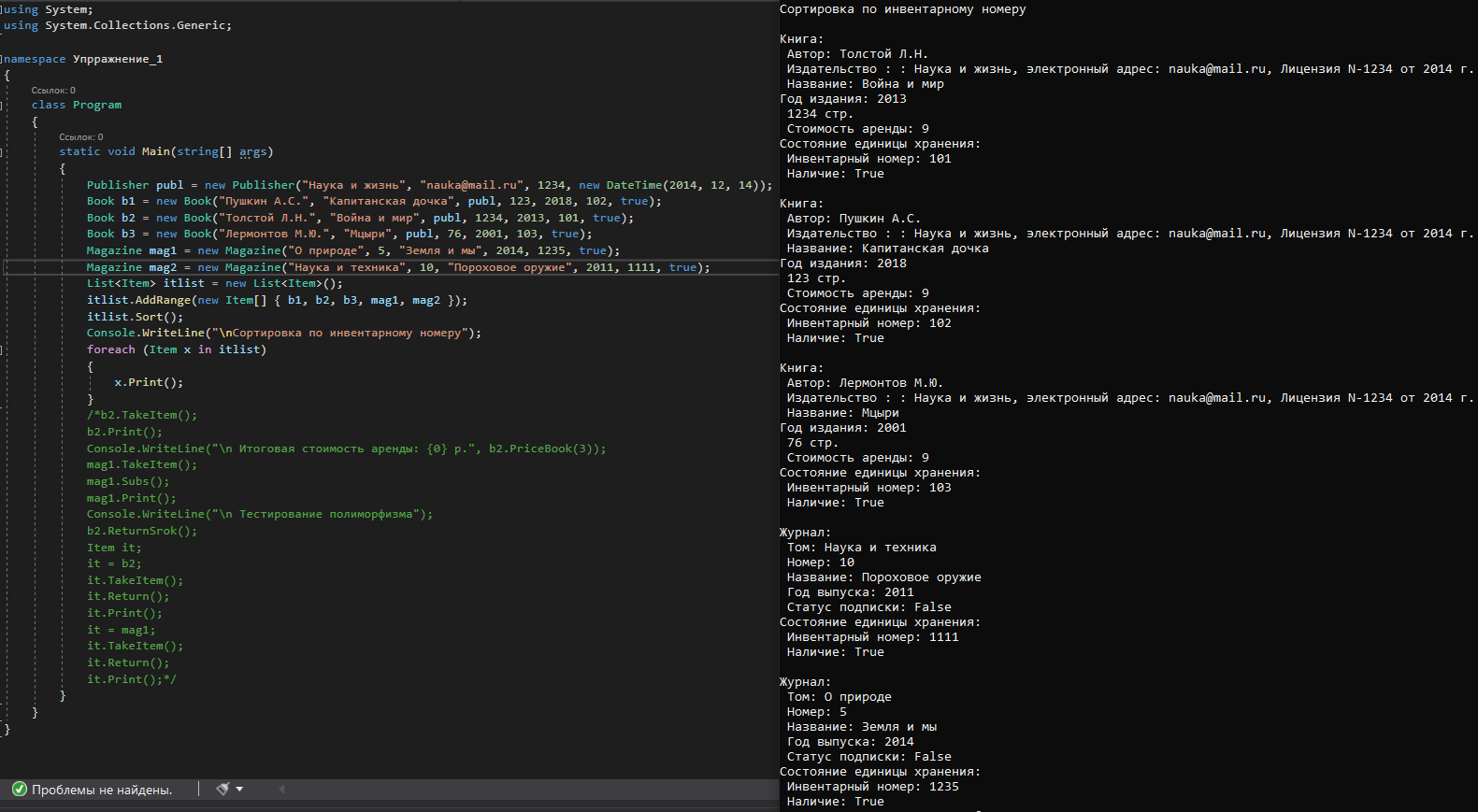


Рисунок 7 – Реализация сортировки по инвентарному номеру

**Упражнение 3.** Определение и использование событий.

В файле Magazine.cs перед классом Magazine объявлен делегат ProcessMagazineDelegate, определяющий сигнатуру метода-обработчика события типа void, принимающий два параметра: объекты типа Magazine и DateTime. В классе Magazine объявлено private событие Subscribe типа ProcessMagazineDelegate, которому присвоено значение null. В методе вызова подписки Subs указана генерация события Subscribe передающая себя в качестве отправителя события и параметр даты для передачи времени подписки.

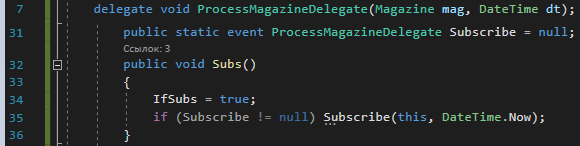


Рисунок 8 – Изменения в классе Magazine

Создан класс Audit задача которого является запись в текстовый файл информации обо всех журналах, на которые оформлена подписка. В него добавлена директива using для пространства имен System.IO. В класс Audit добавлен метод-обработчик события Subscribe MetodSubs, в котором реализована запись информации об оформлении подписки: в блоке try поток вывода перенаправляется в файл “infoSubscribe.txt” c помощью стандартного класса StreamWriter; поток ввода-вывода закрывается с помощью метода Close. Далее идёт блок catch, обрабатывающий стандартные ошибки. Помимо этого, в классе Audit определены статические методы RunAudit, в котором происходит подписка на событие Subscribe, и StopAudit, во котором происходит отказ от подписки на указанное событие.

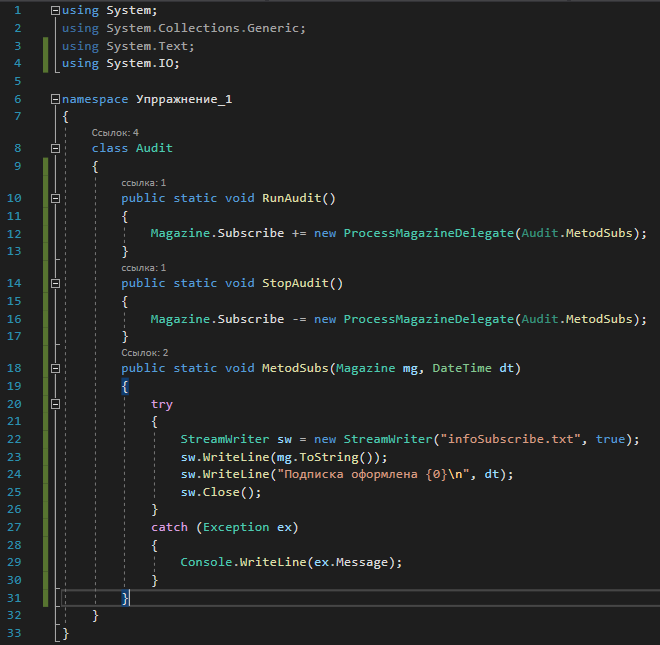


Рисунок 9 – Новый класс Audit

В методе Main класса Program перед созданием объекта mag1 класса Magazine вызван метод RunAudit(). Объект mag1 вызывает метод Subs() и происходит запись об оформленной подписке на журнал. Добавлен метод Audit.StopAudit() перед вызовом mag2.Subs(). Таким образом подписки на второй журнал не происходит.

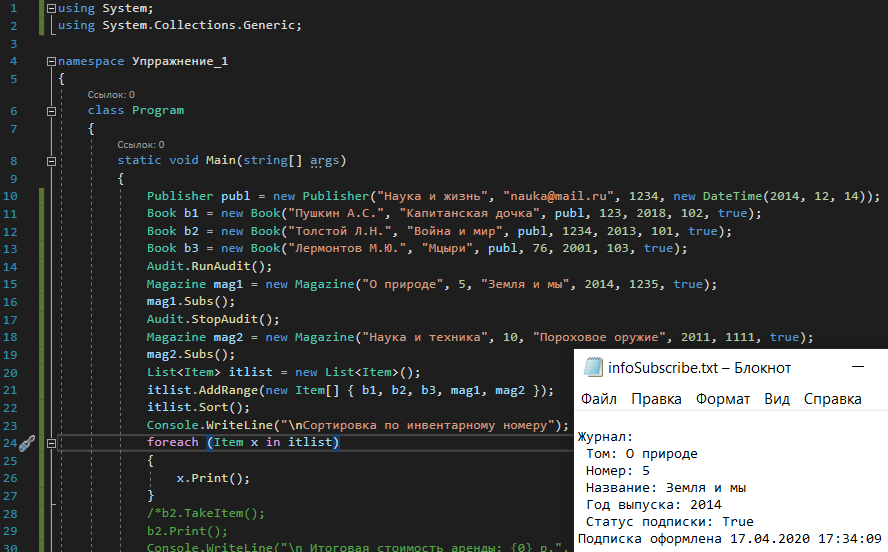


Рисунок 10 – Окончательный код программы

**Упражнение 4.** Реализация прогрессии с помощью интерфейса.

Создан интерфейс IProgression, с одним методом GetElement, принимающим один параметр - номер элемента, который нужно найти. Определены два класса ArithmeticProgression и GeometricProgression, реализующие описанный интерфейс, и описывающие арифметическую и геометрическую прогрессии. В каждом из них определён конструктор, задающий параметры прогрессии и определён унаследованный метод GetElement. В методе Main создаются объекты производных классов с помощью конструкторов и выводятся на консоль значения элементов, указанных пользователем.

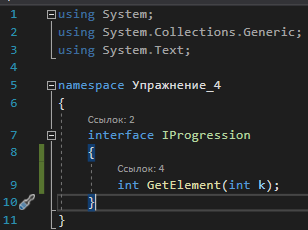


Рисунок 11 – Интерфейс IProgression

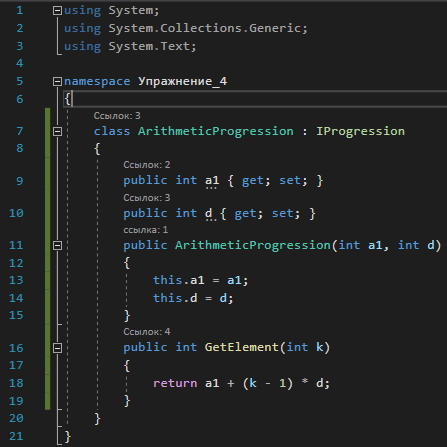


Рисунок 12 – Класс ArithmeticProgression

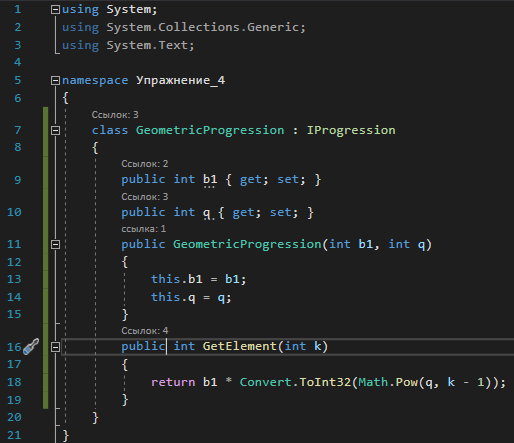


Рисунок 13 – Класс GeometricProgression

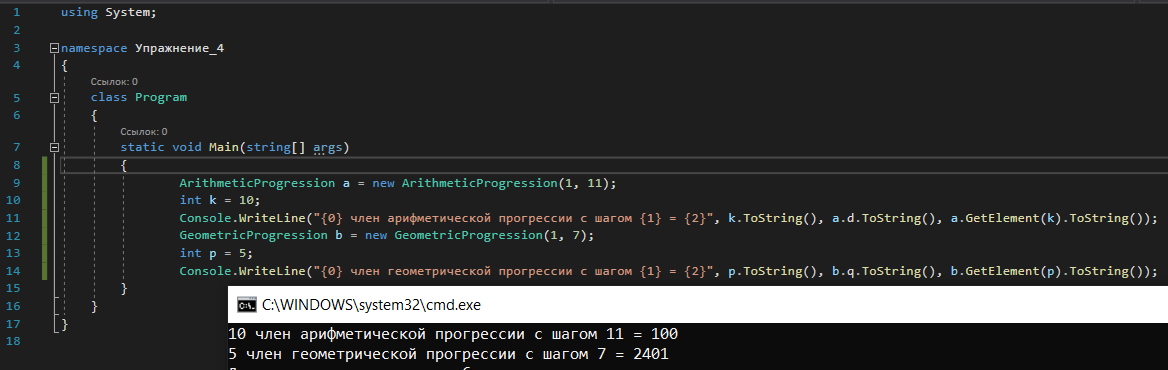


Рисунок 14 – Основная логика программы

**Вывод:**

В результате выполнения лабораторной работы рассмотрено: создание и реализация интерфейсов; определение и использование событий (ключевое слово event) и делегатов (ключевое слово delegate); способ записи данных в файл с помощью стандартного класса StreamWriter; а также пример коллекционного типа данных – списка.