

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

Кафедра программных систем

**Лабораторная работа №7**

**Создание иерархии классов**

Выполнил: Мордвинцев

Максим Витальевич

Группа № K3120

Проверила: Казанова П.П.

Санкт-Петербург

2019

**Цель работы:**

В ходе данной лабораторной работы необходимо изучить иерархию классов.

**Ход работы:**

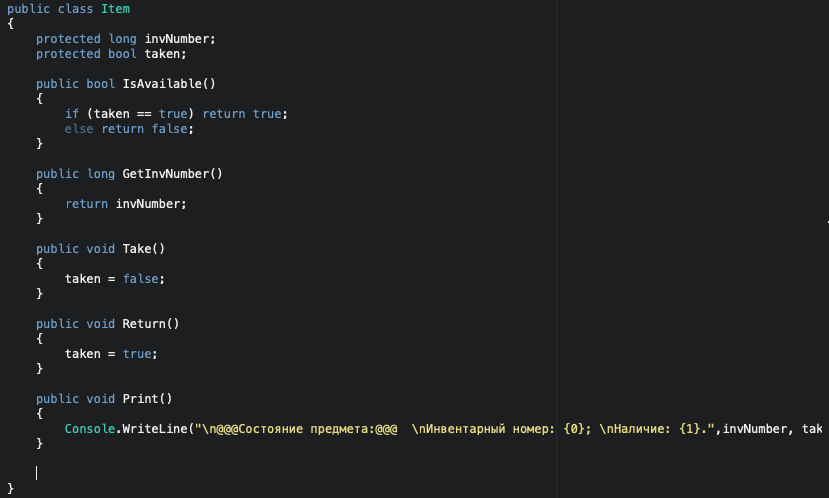
1. Был создан класс Item, где были реализованы основные методы этого класса 

Рисунок 1 – класс Item

1. Класс Book был переделан под производный класс от Item. Также был реализован метод для того, чтобы взять книгу: 

Рисунок 2 – класс Book

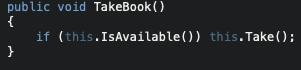


Рисунок 3 – класс book

1. В классе Program был создан объект классе item и программа была протестирована:



Рисунок 4 – вывод программы

1. В класс Item были добавлены конструкторы:

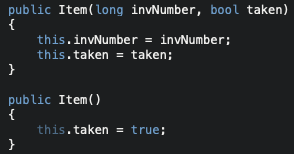


Рисунок 5 - класс item

1. В конструктор Book была добавлена ссылка на базовый класс. Также был дополнен метод вывода информации.

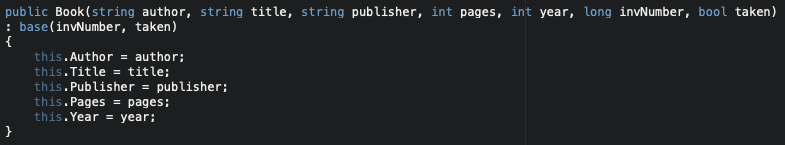


Рисунок 6 – класс book

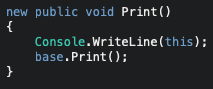


Рисунок 7 - класс book

1. Был добавлен объект класса Book, программа была протестирована.

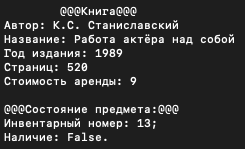


Рисунок 8 – вывод программы

1. Был создан класс Magazine, который является производным от Item. Были определены основные методы и создан конструктор.

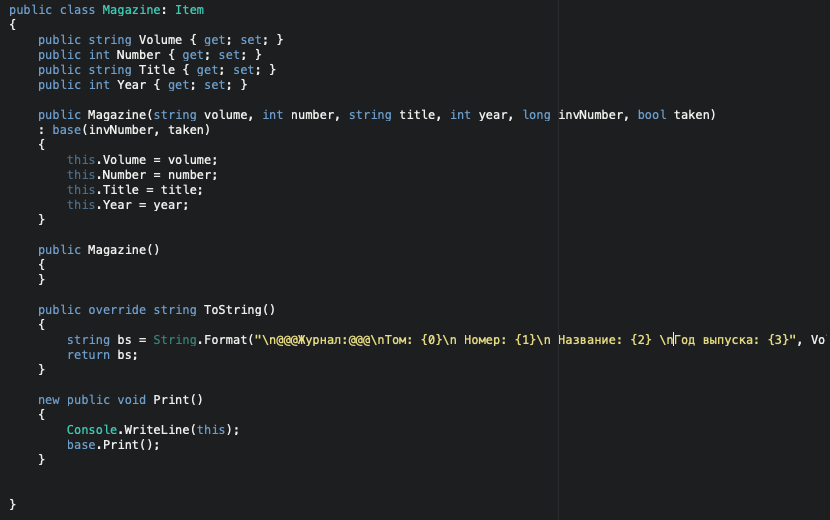


Рисунок 9 – класс Magazine

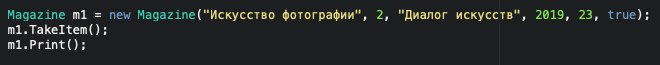
1. В классе Program был создан объект класса Magazine:

Рисунок 10 – класс program

1. Программа была протестирована:

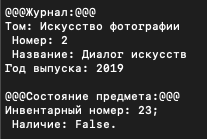


Рисунок 11 – вывод программы

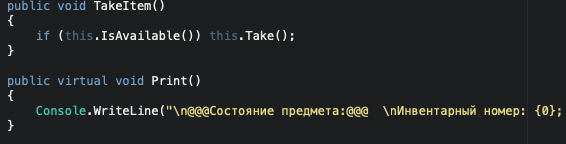
1. В классе Item были созданы виртуальные методы: 

Рисунок 12 – класс Item

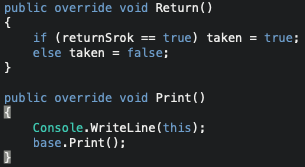
1. В классах Book и Magazine были переопределены эти методы: 

Рисунок 13 – класс Book

1. В классе Program была объявлена ссылка на элемент базового класса, программа была протестирована:

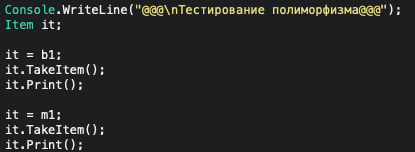


Рисунок 14 – класс program

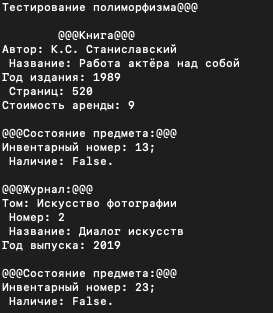


Рисунок 15 – вывод программы

1. Класс Item был реализован как abstract. Также метод Return был определён как abstract:



Рисунок 16 – класс Item

1. Программа была протестирована:

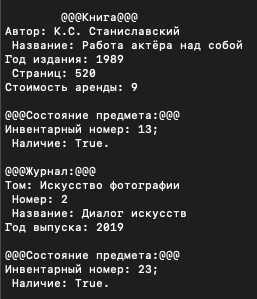


Рисунок 17 – вывод программы

1. Был создан класс Publisher, где были реализованы все его методы.

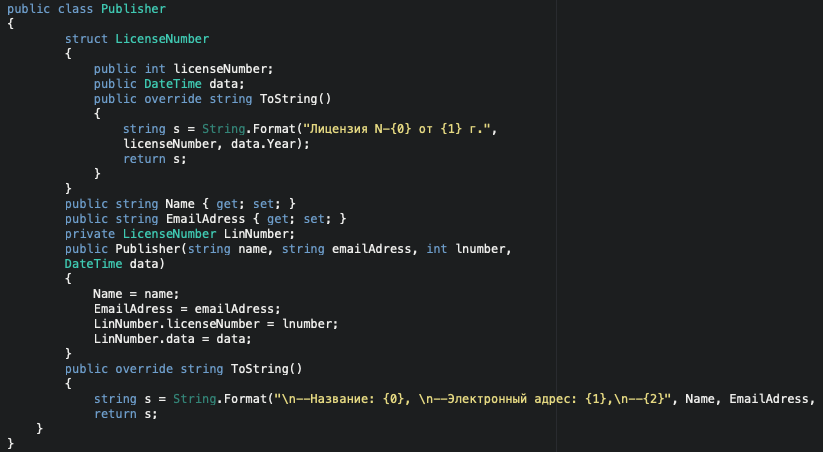


Рисунок 18 – класс publisher

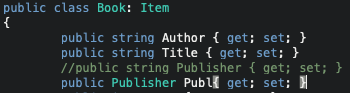
1. Класс Book был изменён, чтобы он принимал объект класса Publisher. 

Рисунок 19 – класс Book



Рисунок 20 – класс book

1. В классе program был создан объект класса Publisher. Программа была протестирована:

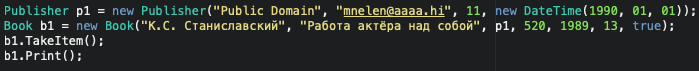


Рисунок 21 – класс program

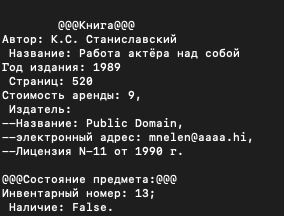


Рисунок 22 – вывод программы

1. Был создан абстрактный класс Progression с методом GetElement.

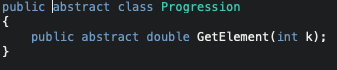


Рисунок 23 – абстрактный класс Progression

1. Были созданы производные Классы ArithmeticProgression и GeometricProgression, где были реализованы методы для расчёта конкретного числа в прогрессии.

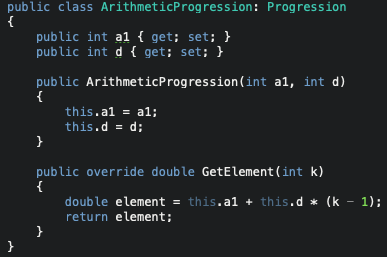


Рисунок 24 – класс ArithmeticProgression

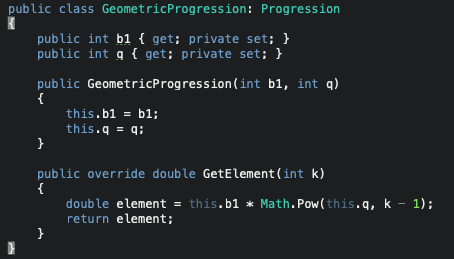


Рисунок 25 – класс GeometricProgression

1. В классе program были созданы два объекта этих классов. Программа была протестирована:

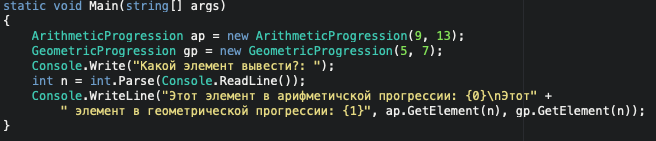


Рисунок 26 – класс program



Рисунок 27 – вывод программы

**Вывод:**

Таким образом, были изучены различные виды отношений между классами, были изучены абстрактные классы и методы, а также виртуальные методы. Полученные знания были применены на практике для создания программы для расчёта числа прогрессии.

**Ответы на контрольные вопросы:**

Напишите здесь контрольные вопросы