МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

кафедра программного обеспечения и администрирования

информационных систем

Отчёт

по лабораторной работе №4

«Наследование»

по дисциплине

###### «Объектно-ориентированные языки и системы»

|  |  |
| --- | --- |
| Выполнил(а): | студент группы 313  Козявин Максим  Сергеевич |
| Проверил: | к.т.н., доцент  кафедры ПОиАИС  Макаров К.С. |

Курск

2023

**Цель работы:** получить практические навыки по применению механизма наследования при создании объектно-ориентированных программ.

**Индивидуальное задание (вариант 4):**

Представьте себе издательскую компанию, которая торгует книгами и аудиозаписями этих книг. Создайте класс Publication, в котором хранятся название (строка) и цена (типа double) книги. Предложим, что издатель решил добавить к своей продукции версии книг на компьютерных дисках для тех, кто любит читать книги на своих компьютерах. Добавьте класс Disk, который является производным класса Publication. Полем только этого класса будет тип диска: CD или DVD. Для хранения этих данных вы можете ввести тип enum. Пользователь должен выбрать подходящий тип, набрав на клавиатуре c или d. Разработайте форму для создания объектов класса Disk, чтобы протестировать возможности ввода/вывода данных.

**Основные теоретические положения**

В языке С# отношения специализации реализуется, как правило, с помощью наследования. Специализация – это отношение «является» (чем-то или кем-то). Пример из задания: Дисковое издание книги является изданием определённой публикации (текста) т.е. класс диска наследуется от класса публикации перенимая все его свойства.

**Экспериментальные результаты**

Код программы для решения задачи:

// Program.cs

internal class Program

{

private static void Main(string[] args)

{

List<Publication> publications = new List<Publication>();

while (true) {

Console.WriteLine("p - publication, d - disk: ");

var publicationType = Console.ReadLine();

if (publicationType == "p" || publicationType == "P") {

Console.WriteLine("Publication name: ");

var name = Console.ReadLine();

Console.WriteLine("Publication price: ");

double price = Convert.ToDouble(Console.ReadLine());

if (name != null) {

publications.Add(new Publication(name, price));

} else {

Console.WriteLine("Wrong input, try again.");

continue;

}

} else if (publicationType == "d" || publicationType == "D") {

Console.WriteLine("Disk name: ");

var name = Console.ReadLine();

Console.WriteLine("Disk type: ");

var diskTypeString = Console.ReadLine();

DiskType diskType;

if (diskTypeString == "CD") {

diskType = DiskType.CD;

} else if (diskTypeString == "DVD") {

diskType = DiskType.DVD;

} else {

Console.WriteLine("Wrong input, try again.");

continue;

}

Console.WriteLine("Disk price: ");

double price = Convert.ToDouble(Console.ReadLine());

if (name != null) {

publications.Add(new Disk(name, price, diskType));

} else {

Console.WriteLine("Wrong input, try again.");

continue;

}

} else {

Console.WriteLine("Wrong input, try again.");

}

foreach (Publication publ in publications) {

Console.WriteLine(publ);

}

}

}

}

// Publication.cs

class Publication {

protected string name;

protected double price;

public Publication(string name, double price) {

this.name = name;

this.price = price;

}

public string Name {

get {

return this.name;

}

}

public double Price {

get {

return this.price;

}

}

public override string ToString() {

return $"{name}, Book, {price}";

}

}

// Disk.cs

enum DiskType {CD, DVD};

class Disk : Publication {

protected DiskType diskType;

public Disk(string name, double price, DiskType diskType) : base(name, price) {

this.diskType = diskType;

}

public DiskType DiskType {

get {

return this.diskType;

}

}

public override string ToString() {

return $"{name}, {diskType}, {price}";

}

}

UML диаграмма:

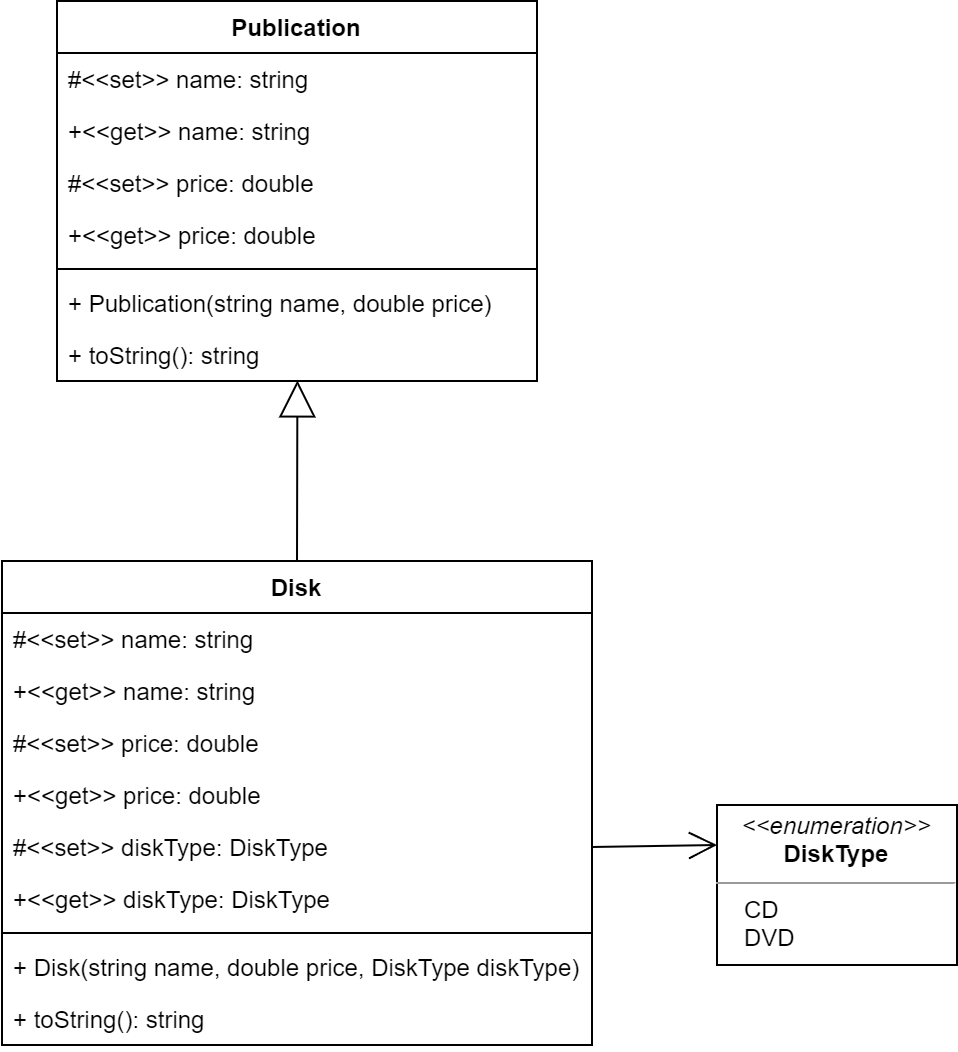


Рисунок 1 – UML диаграмма классов

Тестирование:

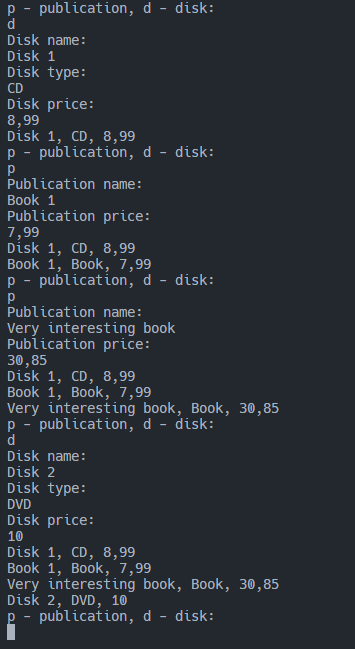


Рисунок 2 – Тест 1 задачи 1

**Вывод:** получил практические навыки по применению механизма наследования при создании объектно-ориентированных программ и научился составлять UML диаграммы классов.