Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №6

на тему:

**«наследование»**

БГУИР 6-05-0612-02 05

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 5.** Предметная область: Товары. Выделить в предметной области 2-3 варианта сущности, отличающиеся несколькими полями и методами. Каждый класс имеет поля, свойства и методы. Спроектировать UML-диаграммы классов. Продемонстрировать работу всех объявленных методов.

# 2 Выполнение работы

В рамках заданной предметной области программа содержит несколько классов: родительский класс Product, производные классы Book и Clothing, производный класс Food, закрытый для наследования. Диаграмма данных классов представлена на рисунке 1.

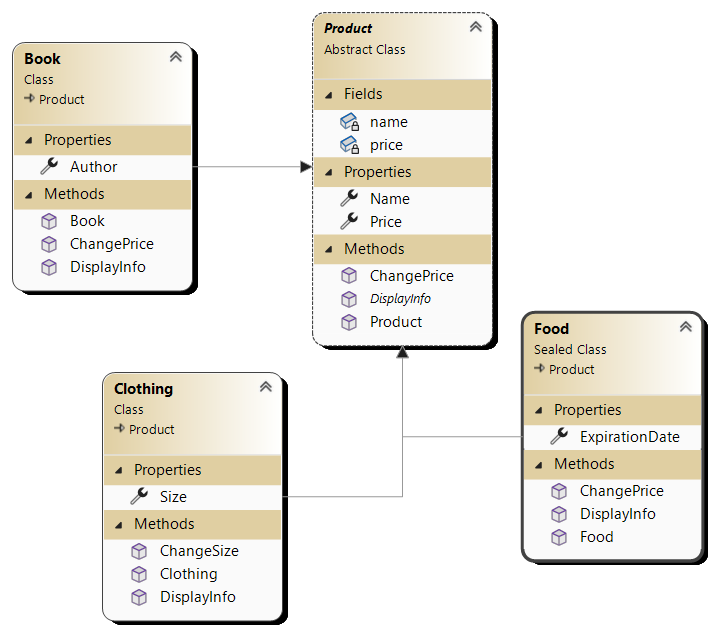


Рисунок 1 – Диаграмма классов

На диаграмме представлен класс Product, который является базовым абстрактным классом. Классы Book и Clothing наследуются от базового класса. Класс Food объявляется с использованием модификатора Sealed, который показывает на то, что класс закрыт для наследования и не может использоваться в качестве базового.

Класс Product является абстрактным базовым классом, который содержит два поля: Name и Price. Эти поля имеют свойства для получения и изменения их значений. В классе Product также определены два метода: абстрактный метод DisplayInfo() и виртуальный метод ChangePrice(). Ниже представлен листинг кода файла Product.cs.

using System;

namespace ConsoleApp1

{

public abstract class Product

{

private string name;

private double price;

public string Name

{

get { return name; }

set

{

if (string.IsNullOrEmpty(value))

throw new ArgumentException("Name cannot be null or empty.");

name = value;

}

}

public double Price

{

get { return price; }

set

{

if (value < 0)

throw new ArgumentException("Price cannot be negative.");

price = value;

}

}

public Product(string name, double price)

{

Name = name;

Price = price;

}

public abstract void DisplayInfo();

public virtual void ChangePrice(double newPrice)

{

Price = newPrice;

}

}

}

Класс Food наследуется от класса Product и добавляет поле ExpirationDate, которое хранит срок годности продукта. В классе Food переопределены методы DisplayInfo() и ChangePrice(). Описан конструктор класса Food для демонстрации вызова конструктора родительского класса при наследовании. Данный класс не доступен для наследования. Ниже представлен листинг кода файла Food.cs.

using System;

namespace ConsoleApp1

{

public sealed class Food : Product

{

public string ExpirationDate { get; set; } // Срок годности

public Food(string name, double price, string expirationDate) : base(name, price)

{

ExpirationDate = expirationDate;

}

public override void DisplayInfo()

{

Console.WriteLine($"Food: {Name}, Price: {Price}, Expiration Date: {ExpirationDate}");

}

public override void ChangePrice(double newPrice)

{

base.ChangePrice(newPrice);

Console.WriteLine($"New price for {Name} is {Price}");

}

}

}

Класс Book также наследуется от класса Product и добавляет поле Author, которое хранит автора книги. В классе Book переопределены методы DisplayInfo() и ChangePrice(). Также имеется метод DisplayInfo(), объявленный с помощью ключевого слова new, который скрывает базовый метод. Ниже представлен листинг кода файла Book.cs.

using System;

namespace ConsoleApp1

{

public class Book : Product

{

public string Author { get; set; }

public Book(string name, double price, string author) : base(name, price)

{

Author = author;

}

public override void DisplayInfo()

{

Console.WriteLine($"Book: {Name}, Price: {Price}");

}

public new void DisplayInfoWithAuthor()

{

Console.WriteLine($"Book: {Name}, Price: {Price}, Author: {Author}");

}

public override void ChangePrice(double newPrice)

{

base.ChangePrice(newPrice);

Console.WriteLine($"New price for {Name} is {Price}");

}

}

}

Класс Clothing наследуется от класса Product и добавляет поле Size, которое хранит размер одежды. В классе Clothing переопределен метод DisplayInfo(), а также добавлен новый метод ChangeSize(). Ниже представлен листинг кода файла Clothing.cs.

using System;

namespace ConsoleApp1

{

public class Clothing : Product

{

public string Size { get; set; }

public Clothing(string name, double price, string size) : base(name, price)

{

Size = size;

}

public override void DisplayInfo()

{

Console.WriteLine($"Clothing: {Name}, Price: {Price}, Size: {Size}");

}

public void ChangeSize(string newSize)

{

Size = newSize;

Console.WriteLine($"New size for {Name} is {Size}");

}

}

}

В главной функции программы создаются объекты каждого из этих классов, и для каждого объекта демонстрируется работа методов DisplayInfo(), ChangePrice() и ChangeSize() (для объекта класса Clothing). Ниже представлен листинг кода файла Program.cs.

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Product[] products = new Product[3];

products[0] = new Food("Apple", 1.2, "2024-13-06");

products[1] = new Book("War and Peace", 12.5, "Leo Tolstoy");

products[2] = new Clothing("T-Shirt", 20.0, "M");

while (true)

{

Console.WriteLine("1. Display product info");

Console.WriteLine("2. Change product price");

Console.WriteLine("3. Change clothing size");

Console.WriteLine("4. Exit");

Console.Write("Enter your choice: ");

string input = Console.ReadLine();

int choice;

if (!int.TryParse(input, out choice) || choice < 1 || choice > 4)

{

Console.WriteLine("Invalid choice. Please enter a number between 1 and 4.");

continue;

}

if (choice == 4)

break;

Console.Write("Enter product index (0-2): ");

input = Console.ReadLine();

int index;

if (!int.TryParse(input, out index) || index < 0 || index > 2)

{

Console.WriteLine("Invalid index. Please enter a number between 0 and 2.");

continue;

}

switch (choice)

{

case 1:

products[index].DisplayInfo();

break;

case 2:

Console.Write("Enter new price: ");

input = Console.ReadLine();

double newPrice;

if (!double.TryParse(input, out newPrice) || newPrice < 0)

{

Console.WriteLine("Invalid price. Please enter a positive number.");

continue;

}

products[index].ChangePrice(newPrice);

break;

case 3:

if (products[index] is Clothing clothing)

{

Console.Write("Enter new size: ");

string newSize = Console.ReadLine();

if (string.IsNullOrEmpty(newSize))

{

Console.WriteLine("Invalid size. Please enter a non-empty string.");

continue;

}

clothing.ChangeSize(newSize);

}

else

{

Console.WriteLine("This product is not a clothing item.");

}

break;

}

}

}

}

}

В программе также имеются классы для проверки на ввод входных данных.

# Вывод

В ходе лабораторной работы были изучены основные принципы, связанные с использованием наследования классов, были изучены возможности создания иерархии классов с общими свойствами и методами, освоены все возможности классов с модификаторами abstract и sealed, а также приобретены навыки использования абстрактных и виртуальных методов.