

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра Информатики
Дисциплина «Программирование»

ОТЧЕТ

к лабораторной работе №8

на тему:

«ПОЛИМОРФИЗМ.»

БГУИР 6-05-0612-02 17

Выполнил студент группы 353502
ХАРИТОНЧИК Денис Сергеевич

(дата, подпись студента)

Проверил ассистент каф.
Информатики
РОМАНЮК Максим Валерьевич

(дата, подпись преподавателя)

Минск 2024

1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Задание 1. Вариант 7. Составить диаграмму классов проектируемой системы. Запрограммировать классы в соответствии с новой диаграммой. Проиллюстрировать использование интерфейсов. Показать вызов метода интерфейса через интерфейсную ссылку. Применить в программе шаблон проектирования Strategy. Предметная область: Фирма грузоперевозок. Фирма имеет список тарифов по перевозке грузов. Класс тариф хранит наименование тарифа и цену. На некоторые тарифы предоставлена скидка, заданная в процентах. В классе фирма реализовать методы добавления нового тарифа и метод поиска тарифа с минимальной стоимостью.

2 ВЫПОЛНЕНИЕ РАБОТЫ

В рамках проекта была создана директория FirmService, в которой размещаются 5 файлов с классами Firm, Tariff, WithDiscountStrategy, WithoutDiscountStrategy, и один интерфейс IPriceCalculationStrategy. Эти классы представляют основные компоненты для системы управления и демонстрации работы с тарифами фирм и стратегиями скидок.

Класс Firm представляет собой класс фирмы. Он хранит приватное поле List<Tariff> для хранения объектов типа Tariff. Метод AddTariff используется для добавления нового тарифа, а метод FindMinPriceTariff для поиска тарифа с минимальной стоимостью. Ниже приведен листинг кода класса Firm.

```
public class Firm
{
    private List<Tariff> _tariffs = new List<Tariff>();

    // метод для добавления тарифа
    public void AddTariff(Tariff tariff)
    {
        _tariffs.Add(tariff);
    }

    // поиск тарифа с минимальной стоимостью
    public Tariff FindMinPriceTariff()
    {
        if (_tariffs.Count == 0)
            return null;
        Tariff minPriceTariff = _tariffs[0];
        foreach (Tariff tariff in _tariffs)
        {
            if (tariff.Price < minPriceTariff.Price)
```

```

        minPriceTariff = tariff;
    }
    return minPriceTariff;
}
}

```

Класс `Tariff` представляет собой класс тарифов. Он хранит три публичные поля и свойства для их чтения и установки. Также в нем содержится конструктор для создания нового тарифа. Ниже приведен листинг кода класса `Tariff`.

```

public class Tariff
{
    public string Name { get; set; }
    public double Price { get; set; }
    public double Discount { get; set; }
    public Tariff(string name, double price, double discount)
    {
        Name = name;
        Price = price;
        Discount = discount;
    }
}

```

Класс `WithDiscountStrategy` создан для расчета стратегии со скидкой. Он реализует интерфейс `IPriceCalculationStrategy` и переопределяет его метод `CalculatePrice`. Реализация метода рассчитывает стоимость с учетом скидки. Ниже приведен листинг кода класса `WithDiscountStrategy`.

```

public class WithDiscountStrategy : IPriceCalculationStrategy
{
    public double CalculatePrice(double originalPrice, double discount)
    {
        return originalPrice * (1 - discount / 100);
    }
}

```

Класс `WithoutDiscountStrategy` создан для расчета стратегии без скидки. Он реализует интерфейс `IPriceCalculationStrategy` и переопределяет его метод `CalculatePrice`. Реализация метода рассчитывает стоимость без учета скидки. Ниже приведен листинг кода класса `WithoutDiscountStrategy`.

```

public class WithoutDiscountStrategy : IPriceCalculationStrategy
{
    public double CalculatePrice(double originalPrice, double discount)
    {
        return originalPrice;
    }
}

```

```

    }
}

```

Интерфейс `IPriceCalculationStrategy` создан для реализации расчета цены. Он хранит метод `CalculatePrice`, который в дальнейшем будет предопределяться в классах, которые будут реализовывать этот интерфейс. Ниже представлен листинг кода интерфейса `IPriceCalculationStrategy`.

```

public interface IPriceCalculationStrategy
{
    double CalculatePrice(double originalPrice, double discount);
}

```

Класс `Program` представляет точку входа в приложение. Внутри метода `Main` создаются объекты классов `Tariff` и `Firm`. Далее выводится минимальный тариф или его отсутствие. После с помощью интерфейса `IPriceCalculationStrategy` и классов, которые выполняют с его контракт создаются объекты и рассчитывается стоимость со скидкой и без. На рисунке 1 изображен вывод данных в консоль. На рисунке 2 изображена UML-диаграмма классов. Ниже представлен листинг кода класса `Program`.

```

class Program
{
    static void Main(string[] args)
    {
        Firm firm = new Firm();

        firm.AddTariff(new Tariff("Стандарт", 100.0, 0.0));
        firm.AddTariff(new Tariff("Примиум", 170.0, 10.0));
        Tariff minPriceTariff = firm.FindMinPriceTariff();
        if (minPriceTariff != null)
        {
            Console.WriteLine($"минимальная стоимость тарифа:
{minPriceTariff.Price}");
        }
        else
        {
            Console.WriteLine("Тарафы не найдены.");
        }

        IPriceCalculationStrategy withDiscountStrategy = new
WithDiscountStrategy();
        double priceWithDiscount = withDiscountStrategy.CalculatePrice(100.0,
10.0);
        Console.WriteLine($"Цена со скидкой: {priceWithDiscount} рублей");

        IPriceCalculationStrategy withoutDiscountStrategy = new
WithoutDiscountStrategy();
        double priceWithoutDiscount =
withoutDiscountStrategy.CalculatePrice(100.0, 10.0);
    }
}

```

```

        Console.WriteLine($"Цена без скидки: {priceWithoutDiscount} рублей");
    }
}

```

```

/Users/denisharitonciksergeevic/RiderProjects/Lab_8/Lab_8/bin/Debug/net6.0/Lab_8
минимальная стоимость тарифа: 100
Цена со скидкой: 90 рублей
Цена без скидки: 100 рублей

Process finished with exit code 0.

```

Рисунок 1 – Вывод данных в консоль

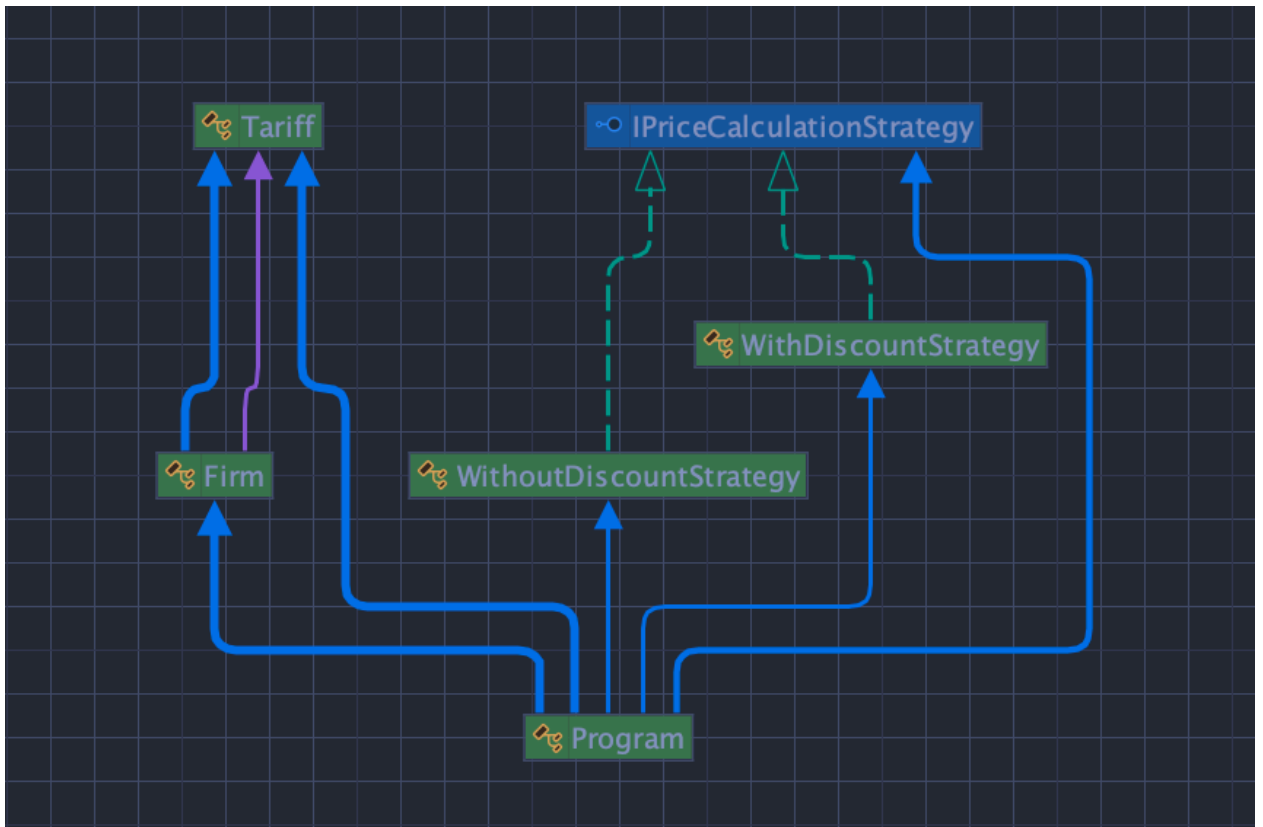


Рисунок 2 – UML-диаграмма классов

ВЫВОД

В ходе лабораторной работы был изучен механизм реализации полиморфизма в C#. Произошло ознакомление с основными подходами при реализации интерфейсов. Получен опыт применения шаблона

проектирование Strategy. Изучено проектирование классов по созданным диаграммам.