

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

Факультет компьютерного проектирования  
Кафедра Информатики  
Дисциплина «Программирование»

## ОТЧЕТ

к лабораторной работе №5

на тему:

**«ИСПОЛЬЗОВАНИЕ КОЛЛЕКЦИЙ.»**

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.** Предметная область: Фирма грузоперевозок. Фирма имеет список тарифов по перевозке грузов. Клиент регистрируется в системе, после чего может заказать перевозку определенного объема груза. Система должна позволять выполнять следующие задачи: ввод тарифов; регистрация клиента и заказ на перевозку грузов; вывод суммы заказа для определенного клиента; подсчет суммарной стоимости всех заказов.

## 2 ВЫПОЛНЕНИЕ РАБОТЫ

В рамках проекта была создана директория Service, в которой размещаются 4 файла с классами Client, Tariff, Order и Firm. Эти классы представляют основные компоненты для системы управления заказов клиентов.

Класс Firm является компонентом для работы программы. Он обладает рядом функциональных возможностей, которые обеспечивают эффективное управление данными и операциями компании. Одной из ключевых особенностей класса является его способность регистрировать клиентов и добавлять тарифы. Это позволяет легко получать доступ к этой информации из различных частей системы, что упрощает её использование. Кроме того, класс предоставляет метод для добавления заказа клиенту, метод для получения цены тарифа и метод для расчета суммарной стоимости заказов всех клиентов. Это позволяет компании эффективно управлять своими финансами и анализировать ее клиентов. Это позволяет компании быстро реагировать на изменяющиеся цены тарифов и количество заказов от зарегистрировавшихся клиентов. Также в классе мы создали два частных поля, которые представляют собой списки объектов типа Tariff и Client. Это позволит хранить в списке Tariff все доступные тарифы, которые могут быть применены к заказам клиентов. Список типа Client будет содержать все объекты Client. Каждый из этих объектов представляет отдельного клиента с его заказами. В этом списке будут храниться все зарегистрированные клиенты компании. Ниже приведен листинг кода класса Firm.

```

public class Firm
{
    private List<Tariff> tariffs = new List<Tariff>();
    private List<Client> clients = new List<Client>();
    public void AddTariff(Tariff tariff)
    {
        tariffs.Add(tariff);
    }
    public void RegisterClient(Client client)
    {
        clients.Add(client);
    }
    public void AddOrderToClient(Client client, double volume, CargoType
cargoType)
    {
        Order order = new Order(client, volume, cargoType);
        client.AddOrder(order);
    }
    public double CalculateTotalOrderSum()
    {
        double totalSum = 0;
        foreach (var client in clients)
        {
            totalSum += client.CalculateOrderSum();
        }
        return totalSum;
    }
    public double GetPriceOfTariff(CargoType cargoType)
    {
        foreach (var tariff in tariffs)
        {
            if (tariff.CargoType == cargoType)
            {
                return tariff.PricePerInit;
            }
        }
        throw new ArgumentException("No tariff found for the specifed cargo
type");
    }
}

```

Класс предоставляет информацию о тарифе на услугу перевозки определенного типа груза. Он обеспечивает хранение информации о тарифах для различных типов грузов, что легко позволяет настраивать настраивать цены на услуги перевозки в зависимости от типа груза. Также в нем содержится перечисление `CargoType`, которое определяет тип груза, на который установлен данный тариф. Например, это может быть тяжеловесный груз, легковесный груз и так далее. Ниже приведен листинг кода класса `Tariff` и перечисления `CargoType`.

```

public enum CargoType
{
    Heavy,
    Light,
    Fragile
}

public class Tariff
{
    public CargoType CargoType { get; }
    public double PricePerInit { get; }
    public Tariff(CargoType cargoType, double pricePerInit)
    {
        CargoType = cargoType;
        PricePerInit = pricePerInit;
    }
}

```

Класс Order представляет собой заказ на перевозку груза от определенного клиента. В нем реализовано свойство Volume, которое представляет количество груза, который необходимо перевести в заказе. Обычно измеряется в единицах объема, например, кубических метрах или килограммах, но здесь оно не указано. Класс позволяет компании хранить информацию о заказах, включая клиента, объем груза и тип груза. Это позволяет компании эффективно управлять заказами и обеспечивать выполнение заказов в соответствии с их требованиями. Ниже приведен листинг кода класса Order.

```

public class Order
{
    public Client Client { get; }
    public double Volume { get; }
    public CargoType CargoType { get; }
    public Order(Client client, double volume, CargoType cargoType)
    {
        Client = client;
        Volume = volume;
        CargoType = cargoType;
    }
}

```

Класс Client представляет собой клиента транспортной компании, который может размещать заказы на перевозку грузов. В нем содержится строковое свойство Name, которое содержит имя клиента. Orders – это список

заказов клиента. Каждый заказ представлен типом объекта Order. Этот список позволяет хранить все заказы, сделанные клиентом. Класс Client обеспечивает возможность хранения о клиентах и их заказах, а также предоставляет методы для работы с этой информацией. Ниже приведен листинг кода класса Client.

```
public class Client
{
    public string Name { get; }
    public List<Order> Orders { get; } = new List<Order>();
    public Client(string name)
    {
        Name = name;
    }
    public void AddOrder(Order order)
    {
        Orders.Add(order);
    }
    public double CalculateOrderSum()
    {
        double sum = 0;
        foreach (var order in Orders)
        {
            sum += order.Volume *
Program.Firm.GetPriceOfTariff(order.CargoType);
        }
        return sum;
    }
}
```

Класс Program представляет точку входа в приложение. Внутри метода Main создаются объекты класса Firm, который управляет работой транспортной компании. Для начала инициализируются тарифы и объекты класса Tariff с помощью метода AddTariff. После этого инициализируются клиенты и регистрируются в фирме с помощью метода RegisterClient. Далее с помощью метода AddOrderToClient заказы добавляются каждому из клиентов. После этого происходит вывод суммы заказов для каждого клиента с помощью метода CalculateOrderSum и вывод суммарной стоимости всех заказов с помощью метода CalculateTotalOrdersSum.

Этот класс обеспечивает инициализацию и управление работой транспортной компании, включая добавление тарифов, регистрацию клиентов, добавление заказов и вывод информации о заказах и их стоимости. Ниже приведен листинг кода класса Program. На рисунке 1 изображен вывод данных в консоль после работы программы. На рисунке 2 изображена UML-диаграмма классов.

```

class Program
{
    public static Firm Firm { get; } = new Firm();
    static void Main(string[] args)
    {
        Firm.AddTariff(new Tariff(CargoType.Heavy, 10));
        Firm.AddTariff(new Tariff(CargoType.Fragile, 15));
        Firm.AddTariff(new Tariff(CargoType.Light, 5));

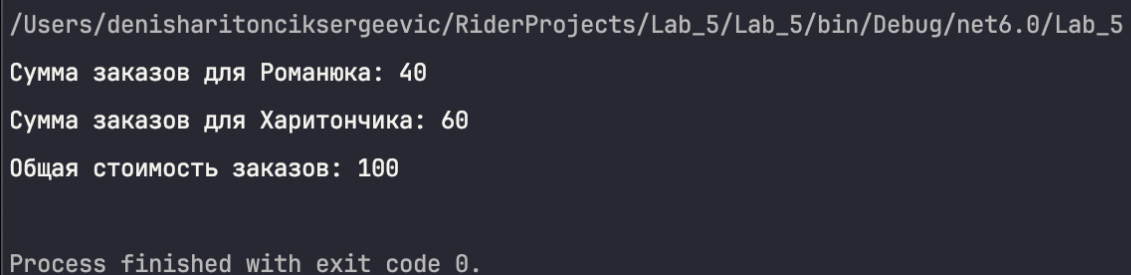
        Client client1 = new Client("Романюк");
        Client client2 = new Client("Харитончик");
        Firm.RegisterClient(client1);
        Firm.RegisterClient(client2);

        Firm.AddOrderToClient(client1, 3, CargoType.Heavy);
        Firm.AddOrderToClient(client2, 4, CargoType.Fragile);
        Firm.AddOrderToClient(client1, 2, CargoType.Light);

        Console.WriteLine("Сумма заказов для Романюка: " +
client1.CalculateOrderSum());
        Console.WriteLine("Сумма заказов для Харитончика: " +
client2.CalculateOrderSum());

        Console.WriteLine("Общая стоимость заказов: " +
Firm.CalculateTotalOrderSum());
    }
}

```



```

/Users/denisharitonciksergeevic/RiderProjects/Lab_5/Lab_5/bin/Debug/net6.0/Lab_5
Сумма заказов для Романюка: 40
Сумма заказов для Харитончика: 60
Общая стоимость заказов: 100

Process finished with exit code 0.

```

Рисунок 1 – Процесс работы консольного приложения

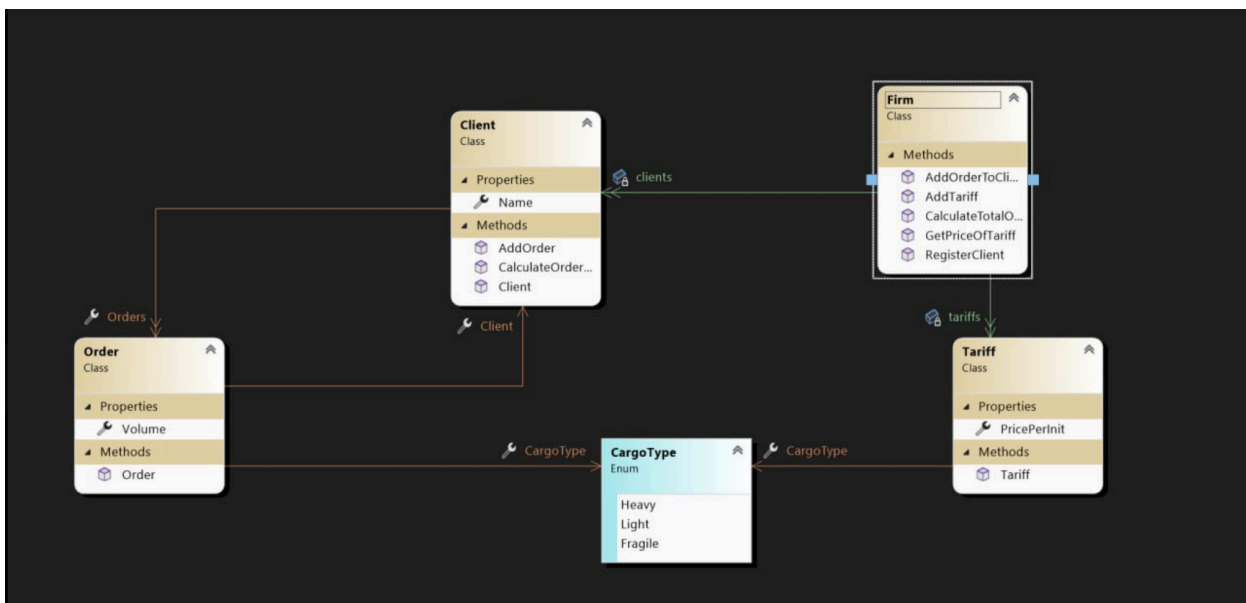


Рисунок 2 – UML-диаграмма классов

## ВЫВОД

В ходе выполнения лабораторной работы были изучены основные принципы создания и использования классов в языке программирования C#. Особое внимание было уделено разработке классов и программ с использованием коллекций и проектированию приложений состоящих из нескольких взаимосвязанных классов. Также была изучен и использован тип данных перечисление.

