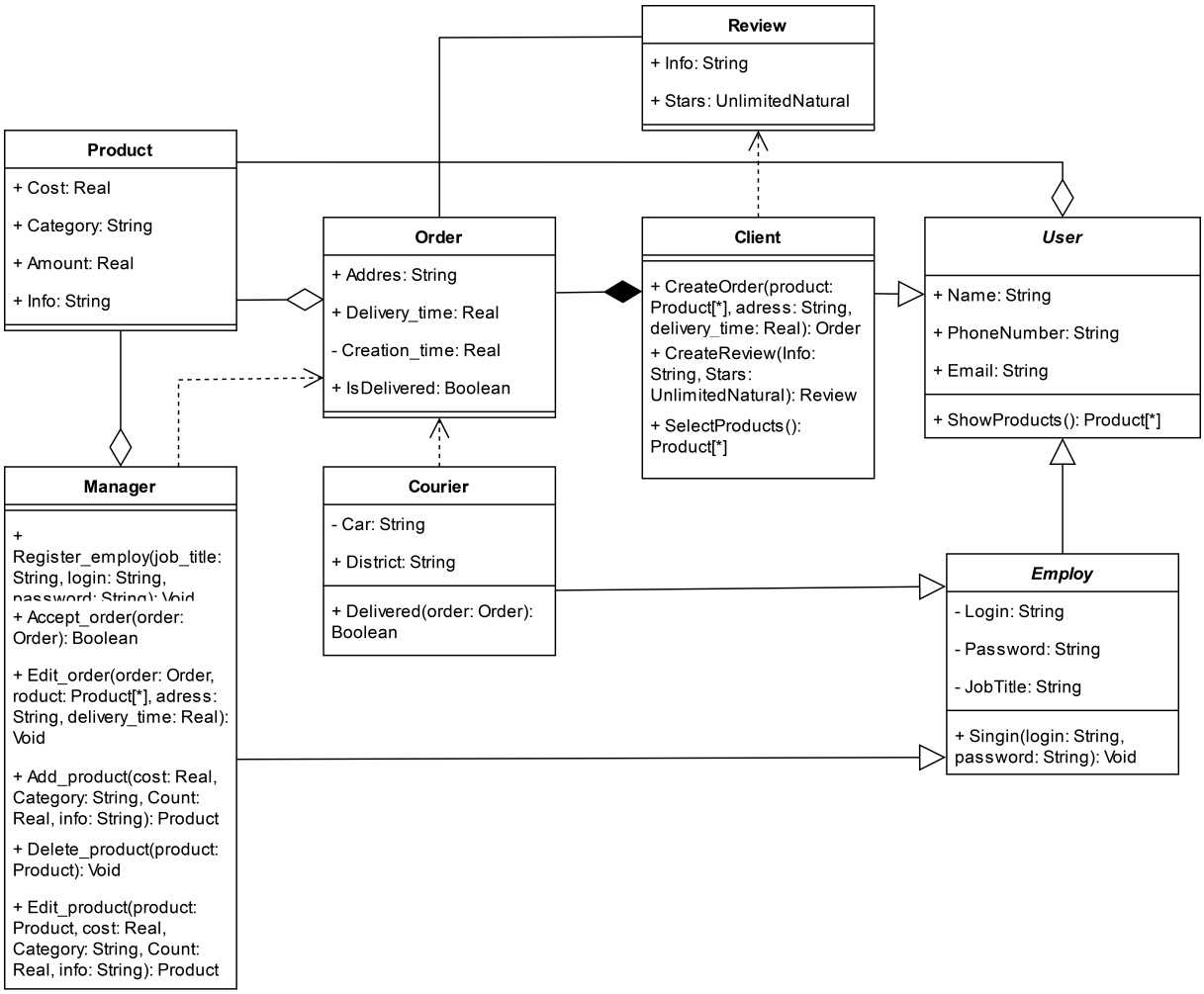
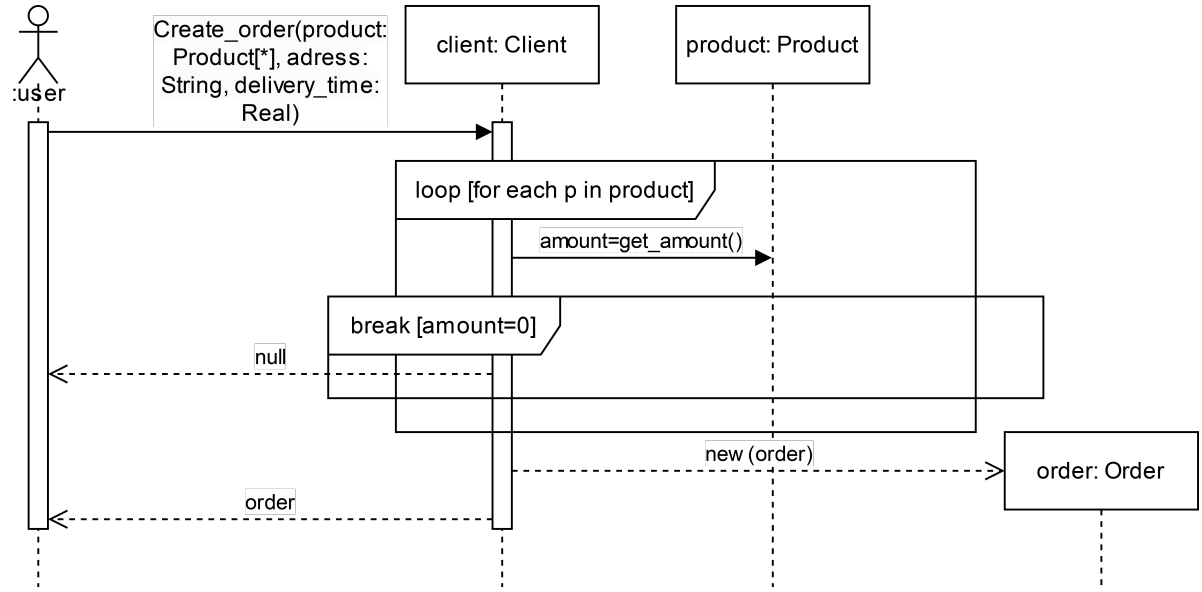
1. Диаграмма классов



1. **Product (Продукт)** - предназначен, для представления товара, который доступен для покупки;
2. **Order (Заказ)** - класс, который хранит всю нужную информацию для заказа;
3. **Review (Отзыв)** - отзыв заказа;
4. ***User (Пользователь)*** - абстрактный класс, который хранит в себе общую информацию;
5. **Client (Клиент)** - класс, который осуществляет действия клиента;
6. ***Employ (Сотрудник)*** - абстрактный класс, который хранит в себе общую информацию сотрудника;
7. **Manager (Менеджер)** - класс, который осуществляет работу менеджера;
8. **Courier (Курьер)** - класс, который осуществляет работу курьера;

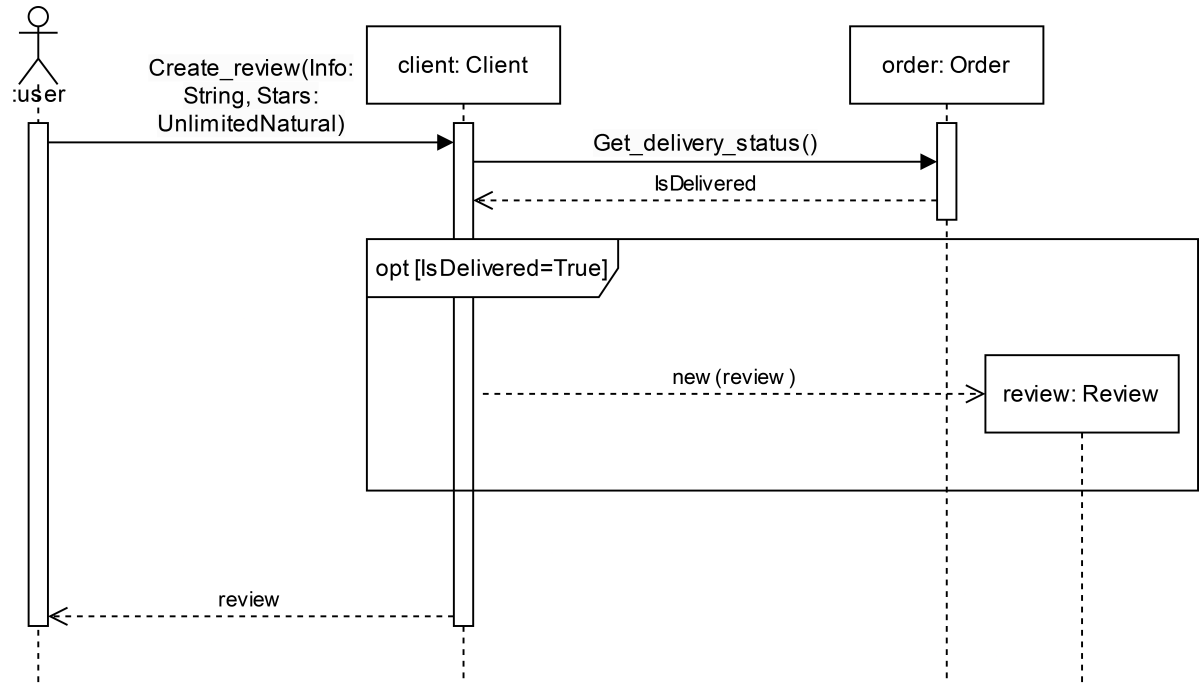
2. Диаграмма последовательностей

2.1 CreateOrder



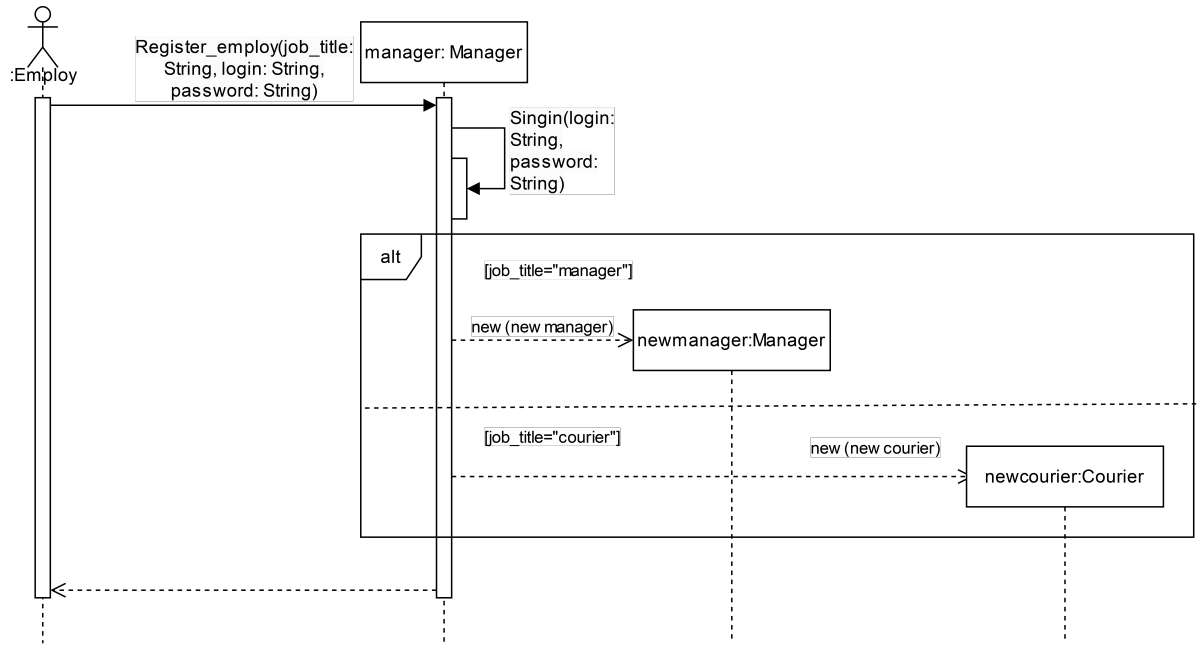
User вызывает метод CreateOrder у объекта client класса Client. Посредством цикла проверяется наличие продукта с помощью использования метода get_amount. Если продукта нет в наличие то заказ не создаётся и происходит выход из цикла посредством break. Если все продукты есть в наличие то будет создан заказ Order

2.2 CreateReview



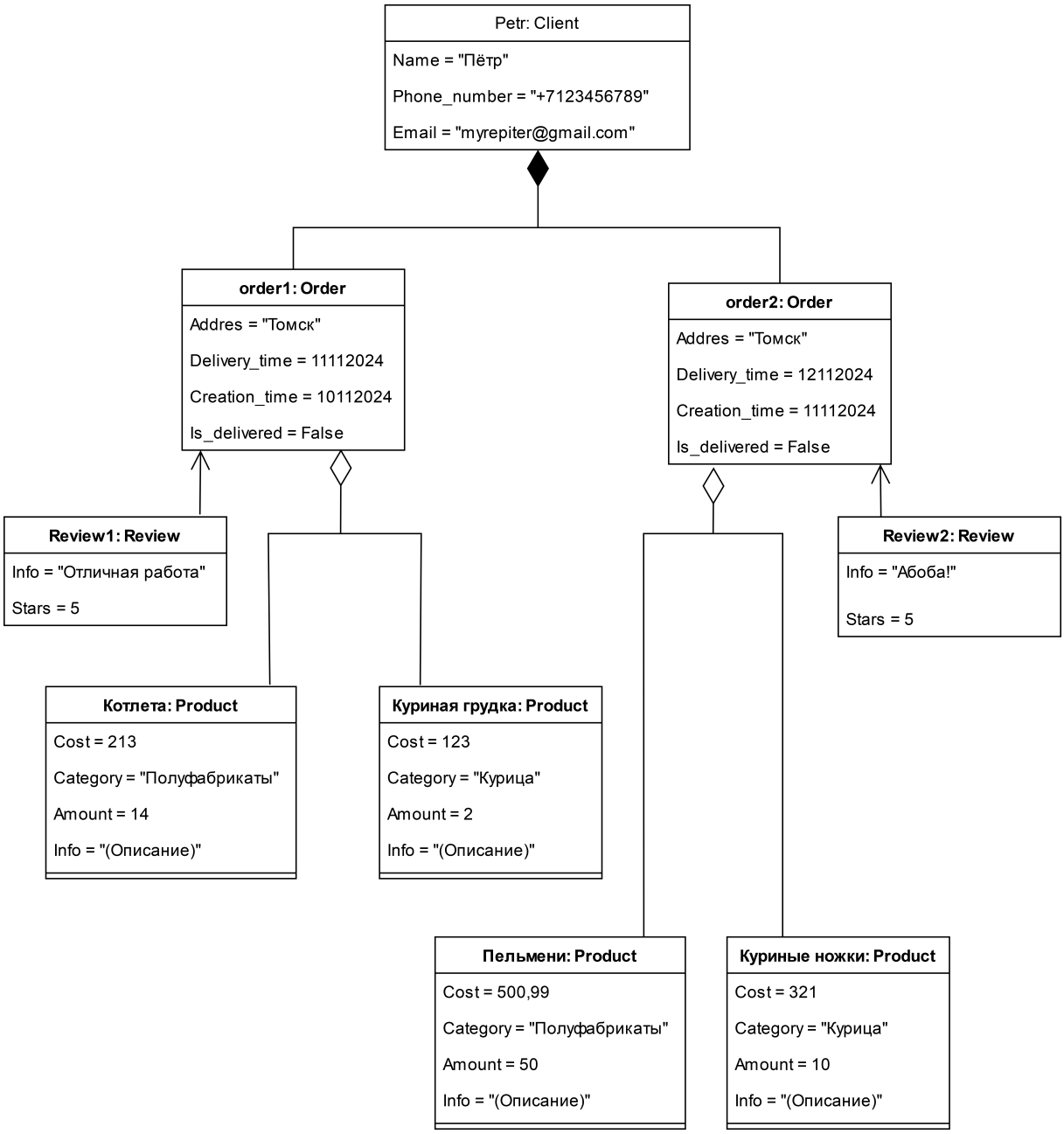
User вызывает метод CreateReview у объекта client класса Client. Получается информация о статусе заказа с помощью метода Get_delivery_status. Если заказ доставлен IsDelivered = True, то новый объект review типа Review

2.3 RegisterEmploy



Employ вызывает метод RegisterEmploy у объекта manager класса Manager. Объекта manager начинает новую активность вызывая метод singin у себя. Если job_title = "manager", то создаётся новый объект newmanager типа Manager, если job_title = "couirier", то создаётся новый объект newcourier типа Courier

3. Диаграмма объектов



У клиента Petr имеется два заказа Order1 и Order2. Order1 имеет отзыв Review1 и два продукта Котлеты и Куринная грудка. Order2 имеет отзыв Review2 и два продукта Пельмени и Куринные ножки

4. Код

```
using System;
using System.Collections.Generic;

namespace OrderManagementSystem
{
    public class Product
    {
        public double Cost { get; set; }
        public string Category { get; set; }
        public double Amount { get; set; }
        public string Info { get; set; }

        public Product(double cost, string category, double amount, string info)
        {
            Cost = cost;
            Category = category;
            Amount = amount;
            Info = info;
        }
    }

    public class Review
    {
        public string Info { get; set; }
        public int Stars { get; set; }

        public Review(string info, int stars)
        {
            Info = info;
            Stars = stars;
        }
    }

    public class User
    {
        public string Name { get; set; }
        public string PhoneNumber { get; set; }
        public string Email { get; set; }
        public List<Product> ShowProducts()
        {
        }
    }

    public class Employ : User
    {
        public string Login { get; set; }
        public string Password { get; set; }
        public string JobTitle { get; set; }

        public void Signin(string login, string password)
        {
        }
    }

    public class Client : User
    {
```

```
    public Order CreateOrder(List<Product> products, string address, double
    ↪  deliveryTime)
    {
        foreach (var product in products)
        {
            if (product.Amount == 0)
            {
                return null;
            }
        }

        return new Order(address, deliveryTime, DateTime.Now, products);
    }

    public void CreateReview(Order order, string info, int stars)
    {
        if (order.IsDelivered)
        {
            order.Review = new Review(info, stars);
        }
        else
        {
            Console.WriteLine("Отзыв можно оставить только после доставки
            ↪  заказа.");
        }
    }

    public List<Product> SelectProducts(){
    }
}

public class Order
{
    public string Address { get; set; }
    public double DeliveryTime { get; set; }
    public double CreationTime { get; set; }
    public bool IsDelivered { get; set; }

    public List<Product> Products { get; set; }
    public Review Review { get; set; }

    public Order(string address, double deliveryTime, double creationTime,
    ↪  List<Product> products)
    {
        Address = address;
        DeliveryTime = deliveryTime;
        CreationTime = creationTime;
        Products = products;
        IsDelivered = false;
        Review = null;
    }
}

public class Manager : Employ
{
    public List<Product> Products { get; private set; }

    public Manager()
```

```
    {
        Products = new List<Product>();
    }

    public void RegisterEmploy(string jobTitle, string login, string password)
    {
        this.Signin(login, password);

        if (jobTitle == "manager")
        {
            Manager newManager = new Manager();
            Console.WriteLine("New manager registered.");
        }
        else if (jobTitle == "courier")
        {
            Courier newCourier = new Courier();
            Console.WriteLine("New courier registered.");
        }
        else
        {
            Console.WriteLine("Invalid job title.");
        }
    }

    public bool AcceptOrder(Order order)
    {
        return true;
    }

    public void EditOrder(Order order, List<Product> products, string address,
    ↪  double deliveryTime)
    {
    }

    public void AddProduct(double cost, string category, double amount, string
    ↪  info)
    {
        Product product = new Product(cost, category, amount, info);
        Products.Add(product);
    }

    public void DeleteProduct(Product product)
    {
        Products.Remove(product);
    }

    public void EditProduct(Product product, double cost, string category, double
    ↪  amount, string info)
    {
        product.Cost = cost;
        product.Category = category;
        product.Amount = amount;
        product.Info = info;
    }
}

public class Courier : Employ
{
```

```
        public string Car { get; set; }
        public string District { get; set; }

        public bool Delivered(Order order)
        {
            order.IsDelivered = true;
            return true;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Manager manager = new Manager();
            manager.RegisterEmploy("Courier", "courier123", "password123");
            manager.AddProduct(500.99, "Полуфабрикат", 50, "Пельмени");

            Client client = new Client();
            client.ShowProducts();
            client.SelectProducts();
            Order order = client.CreateOrder(new List<Product> { manager.Products[0]
            ↪  }, "ул. Ленина 36а", 2400.0);

            manager.AcceptOrder(order);
            Courier courier = new Courier { Car = "a200cx", District = "Кировский" };
            courier.Delivered(order);
            Console.WriteLine($"Заказ доставлен: {order.IsDelivered}");


            client.CreateReview(order, "Аббба!", 5);

            if (order.Review != null)
            {
                Console.WriteLine($"Отзыв к заказу: {order.Review.Info}, звёздочек:
                ↪  {order.Review.Stars}");
            }
        }
    }
}
```