МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Факультет информационных систем и технологий
Кафедра интеллектуальных систем автоматизации и управления

Направление подготовки: 27.03.04 Управление в технических системах

Направленность (профиль): Информационные технологии в управлении

Отчет по лабораторной работе № 1
по дисциплине:
Объектно-ориентированное программирование в управлении техническими системами.
на тему:
Объектно-ориентированный анализ предметной области.

Выполнил студент группы ИСТ-052
Колесников Е. В.
Фамилия И. О.

Проверил к.т.н., доцент
уч. степень, уч. звание
Олимпиев А. А.
Фамилия И. О.

дата, подпись

Санкт-Петербург
2022

Цели работы.

Научиться проводить объектно-ориентированный анализ предметной области, составлять UML-диаграммы классов, написать прецеденты. Разработать библиотеку классов на языке C#

Выделение сущностей.

Я провел анализ предметной области «сфера услуг» и выбрал в качестве примера доставку еды, с возможностью ее выбора в нескольких ресторанах. Были выделены следующие сущности:

- Рестораны (ресторанный эксперт)
- Пункт меню ресторана (блюдо)
- Покупатель
- Заказ (виртуальный официант)

Отношения между возможными сущностями в выбранном примере:

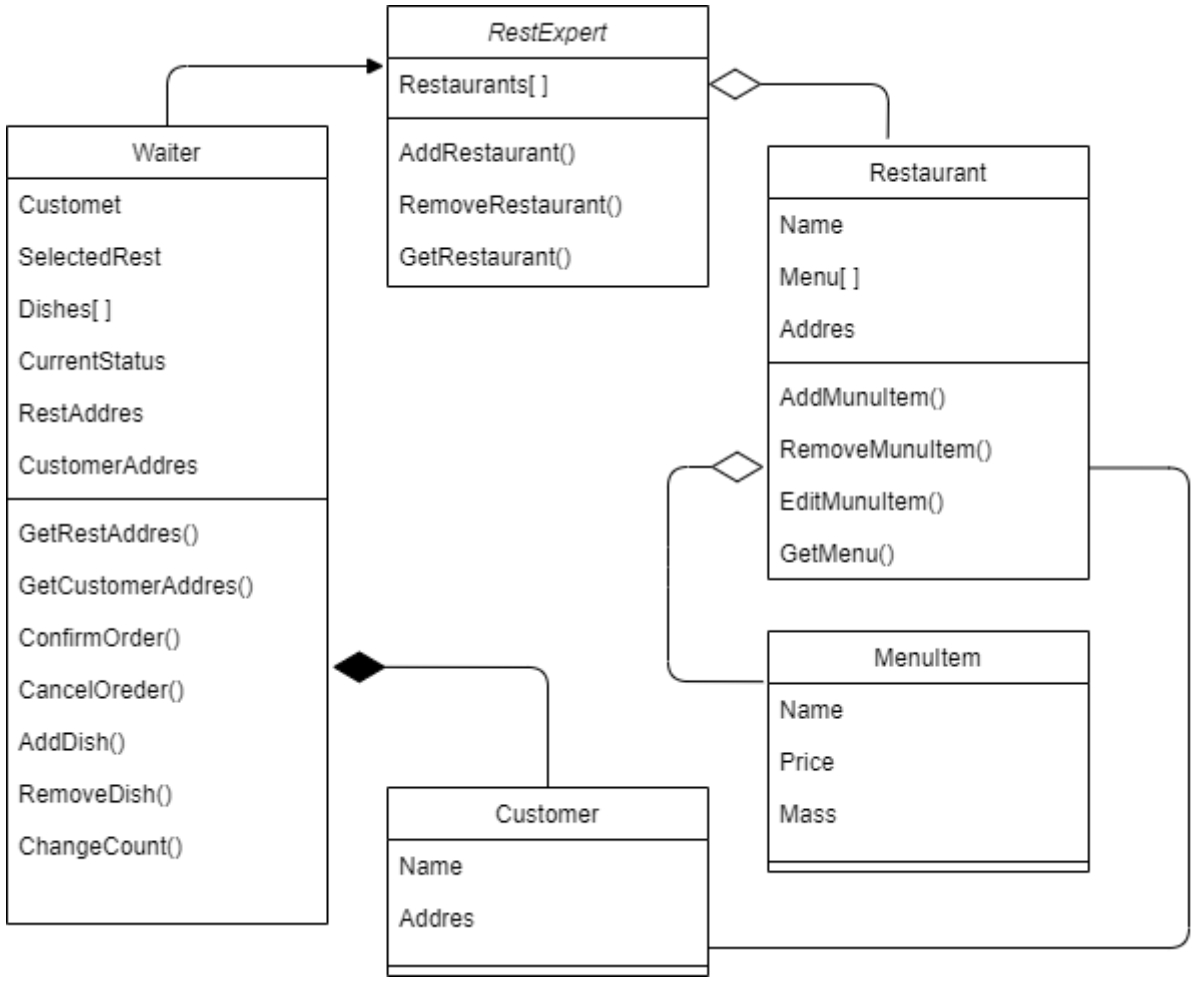
Название отношения	**Сущность 1**	**Сущность 2**	**Описание**
Ассоциация	Покупатель	Ресторан	Покупатель и ресторан – главные сущности, но они не взаимодействуют на прямую.
Наследование	Ресторан	Пиццерия	Пиццерия – это частный случай ресторана: имеет свою концепцию.
Агрегация	Меню (концепция) ресторана	Блюдо	Концепция ресторана подразумевает определённые блюда, но набор этих блюд не строгий, может изменяться в пределах концепции.
Композиция	Ресторан	Меню (концепция) ресторана	Концепция ресторан – его неотъемлемая часть. Смена концепции – смена всех остальных сущностей внутри ресторана.
Взаимодействие	Покупатель	Заказ (корзина)	Покупатель использует заказ (корзину), для того чтобы получить еду на дом.
Зависимость	Заказ	Ресторан	Заказ не может существовать без ресторанов, где его можно сделать.
Реализация	Горожанин	Покупатель	Горожанин играет роль покупателя.


Покупатель
Ресторан
Пиццерия
Ресторан
Блюдо
Меню (концепция) ресторана
Меню (концепция) ресторана
Ресторан
Покупатель
Заказ
Заказ
Блюдо
Горожанин
Покупатель

Итого, в данном примере действующими лицами будут являться: покупатель, ресторан и служба доставки.

- Ресторан заключает договор со службой доставки и предоставляет ей информацию о своем меню и метаположении.
- Покупатель открывает заказ и посредством службы доставки выбирает ресторан.
- Покупатель выбирает блюда из меню избранного ресторана в некотором количестве.
- Покупатель подтверждает заказ, а служба доставки и ресторан получают соответствующую информацию: ресторан – информацию о блюдах в заказе, служба доставки – адрес выбранного ресторана и адрес покупателя.

Диаграмма классов:



Пример использования библиотеки.

Добавление ресторанов в список известных ресторанов и добавление блюд в меню ресторанов:

```
RestExpert Expert = new RestExpert();

Expert.AddRestaurant("ДоДо Пицца", "6 линия В. О.");
Expert.AddRestaurant("Макдональдс", "6 линия В. О.");

Expert.Restaurants["ДоДо Пицца"].AddMenuItem("Пицца \"Двойной цыпленок\"", 400, 299);
Expert.Restaurants["ДоДо Пицца"].AddMenuItem("Картофель фри", 100, 99);
Expert.Restaurants["ДоДо Пицца"].AddMenuItem("Капучино", 350, 99);

Expert.Restaurants["Макдональдс"].AddMenuItem("Биг Тейсти", 400, 319);
Expert.Restaurants["Макдональдс"].AddMenuItem("Картофель фри", 100, 99);
Expert.Restaurants["Макдональдс"].AddMenuItem("Пепси", 350, 99);
```

```
public class RestExpert {

    public Dictionary<string, Restaurant> Restaurants;

    public RestExpert() {
        Restaurants = new Dictionary<string, Restaurant>();
    }

    public void AddRestaurant(string name, string address) {
        Restaurant restaurant = new Restaurant(name, address);
        Restaurants.Add(restaurant.Name, restaurant);
    }

    public void RemoveRestaurant(string name) {
        Restaurants.Remove(name);
    }

    public Restaurant GetRest(string name) {
        return Restaurants[name];
    }
}
```

```
public class Restaurant {

    public Guid ID;
    public string Name;
    public string Address;
    public Dictionary<string, MenuItem> Menu;

    public Restaurant(string name, string address) {
        ID = Guid.NewGuid();
        Name = name;
        Address = address;
        Menu = new Dictionary<string, MenuItem>();
    }

    public void AddMenuItem(string name, int mass, decimal price) {
        Menu.Add(name, new MenuItem(name, mass, price));
    }

    public void RemoveMenuItem(string name) {
        Menu.Remove(name);
    }

    public void EditMenuItem(string name,
        string newName = "", int newMass = 0, decimal newPrice = 0) {

        MenuItem item = Menu[name];
        if (newName != "") { item.Name = newName; }
        if (newMass != 0) { item.Mass = newMass; }
        if (newPrice != 0) { item.Price = newPrice; }
        Menu[name] = item;
    }

    public Dictionary<string, MenuItem> GetMenu() {
        return Menu;
    }
}
```

В работу включается виртуальный официант, формирующий заказ и хранящий информацию о нем:

```
Customer Evgenii = new Customer("Евгений Колесников", "12 линия В. О.");
Waiter MyWaiter = new Waiter(Evgenii, Expert.GetRest("ДоДо Пицца"));
```

```
public class Waiter {

    public Guid ID;
    public Customer Customer;
    public Restaurant SelectedRest;
    public Dictionary<MenuItem, int> Dishes;
    public DateTime OrderTime;
    public enum Status {
        Отменен = 0,
        Оформляется = 1,
        Подтвержден = 2,
    }
    public Status CurrentStatus;

    public Waiter(Customer customer, Restaurant restaurant) {
        ID = Guid.NewGuid();
        Customer = customer;
        SelectedRest = restaurant;
        Dishes = new Dictionary<MenuItem, int>();
        CurrentStatus = Status.Оформляется;
    }
```

Методы виртуального официанта:

```
public string GetCustomerAddress() {
    return Customer.Address;
}

public string GetRestaurantAddress() {
    return SelectedRest.Address;
}

public void ConfirmOrder() {
    CurrentStatus = Status.Подтвержден;
}

public void CancelOrder() {
    CurrentStatus = Status.Отменен;
}

public void AddDish(string MenuItem) {
    var dish = SelectedRest.Menu[MenuItem];
    if (Dishes.ContainsKey(dish)) {
        Dishes[dish] += 1;
        return;
    }
    Dishes.Add(dish, 1);
}

public void RemoveDish(string MenuItem) {
    var dish = SelectedRest.Menu[MenuItem];
    if (Dishes.ContainsKey(dish)) {
        Dishes.Remove(dish);
    }
}

public void ChangeCount(string MenuItem, int count) {
    var dish = SelectedRest.Menu[MenuItem];
    if (Dishes.ContainsKey(dish)) {
        if (count == 0) {
            RemoveDish(MenuItem);
            return;
        }
        Dishes[dish] = count;
    }
}
```

Добавление пунктов заказа:

```
MyWaiter.AddDish("Картофель фри");
MyWaiter.AddDish("Пицца \"Двойной цыпленок\"");
MyWaiter.AddDish("Капучино");

MyWaiter.AddDish("Пицца \"Двойной цыпленок\"");
MyWaiter.ChangeCount("Капучино", 2);
MyWaiter.ChangeCount("Картофель фри", 10);
MyWaiter.ChangeCount("Картофель фри", 2);
```

Вывод.

В ходе работы был проведен объектно-ориентированный анализ предметной области «сфера обслуживания», а именно был проанализирован сервис доставки еды из нескольких ресторанов. По выбранному примеру создана библиотека классов.