****­МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Дальневосточный федеральный университет»**

**ИНСТИТУТ ИНФОРМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Департамент информационных и компьютерных систем**

**ОТЧЕТ**

ПО ИТОГОВОЙ ЛАБОРАТОРНОЙ РАБОТЕ

по дисциплине **«****Объектно-ориентированный анализ и программирование»**

на тему **«Объективно-ориентированный анализ и программирование»**

|  |  |
| --- | --- |
| Выполнил студент гр. | |
| Б9121-09.03.03пиэ(2) | |
|  | В.Ю. Туровец |
|  | |
| Проверил старший преподаватель | |
|  | Е.И. Шувалова |
|  | |
| (оценка) | |

г. Владивосток

2022

Оглавление

[1 Анализ предметной области 3](#_Toc130212759)

[1.2 Сценарии 3](#_Toc130212760)

[1.3 Диаграмма Use Case 3](#_Toc130212761)

[1.4 Список всех существительных 4](#_Toc130212762)

[1.5 Диаграмма классов 5](#_Toc130212763)

[1.6 Диаграмма последовательности 5](#_Toc130212764)

[1.7 Диаграмма коммуникации 6](#_Toc130212765)

[1.8 Диаграмма деятельности 7](#_Toc130212766)

[2 Реализация 8](#_Toc130212767)

[2.1 Описание классов 8](#_Toc130212768)

[3 Организация структуры папок проекта 12](#_Toc130212769)

[Вывод 13](#_Toc130212770)

# 1 Анализ предметной области

Ресторан – это предприятие общественного питания с широким ассортиметом блюд сложного приготовления, включая фирменные. В качестве предпренимательской деятельности ресторан осуществляет приготовление и вадачу заказанных блюд клиенту.

Кто-то посещает ресторан в обеденное время, кто-то вечером, кто-то отмечает в нем самые важные моменты жизни.

Каждый клиент выбирает блюдо по вкусу, поэтому ресторан предлагает достаточно широкий ассортимент. Поскольку утром клиентов приходит мало, режим работы был выбран с 11.00-23.00.

Функциональное назначение: формирование заказа в соответствии с заказом клиента.

Задание: автоматизация работы ресторана по формированию и выдаче заказов.

Ресторан осуществляет процессы по приготовлению и выдаче заказанных блюд.

## 1.2 Сценарии

Сценарий №1 - Клиент может ознакомиться с меню и сделать заказ.

Сценарий №2 - Официант получает заказ от клиента. Далее заказ предаётся на кухню.

Сценарий №3 - После выполнения заказа поваром официант приносит его клиенту. После происходит оплата.

## 1.3 Диаграмма Use Case

С помощью Use Case можно описать взаимодействие участников между друг другом. На рисунке 1 представлена диаграмма вариантов использования ресторана.

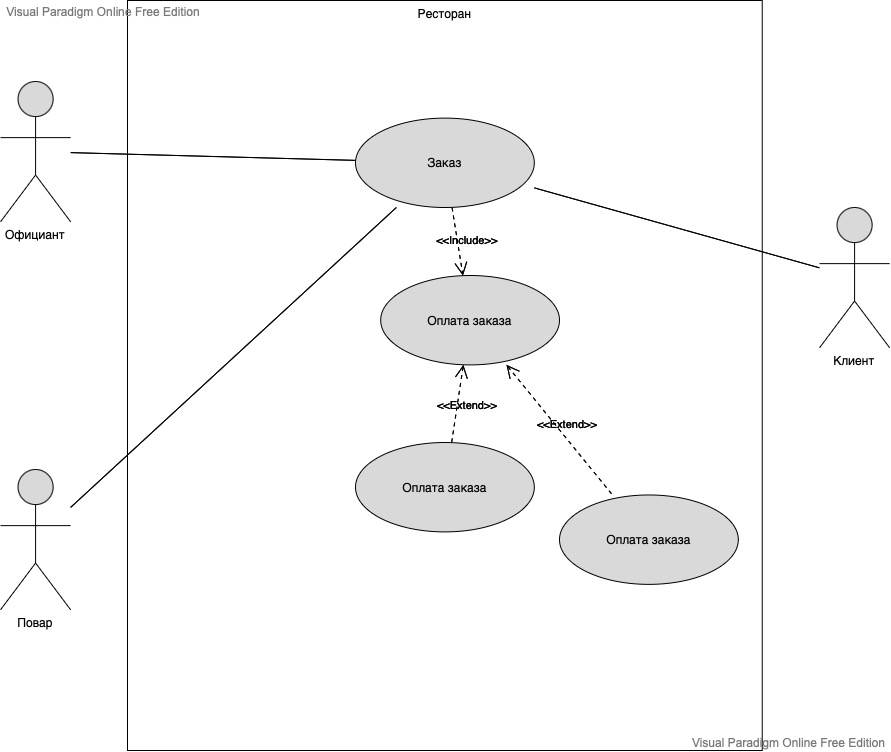


Рисунок 1 – Диаграмма вариантов использования

## 1.4 Список всех существительных

1. Клиент.
2. Официант
3. Повар.
4. ФИО сотрудника.
5. Должность.
6. Меню.
7. Блюда.
8. Напитки.
9. Заказ.
10. Оплата.
11. Номер заказа.
12. Статус заказа.

## 1.5 Диаграмма классов

Выделены следующие классы: работник, повар, официант, клиент, меню, заказ (см. Рисунок 2).

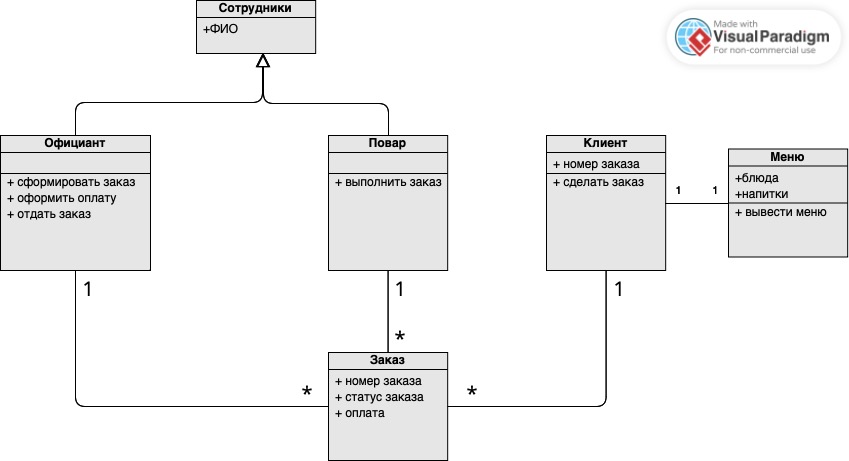


Рисунок 2 – Диаграмма классов

## 1.6 Диаграмма последовательности

На рисунке 3 представлена диаграмма, описывающая все взаимодействия объектов в упорядоченной псоледовательности.

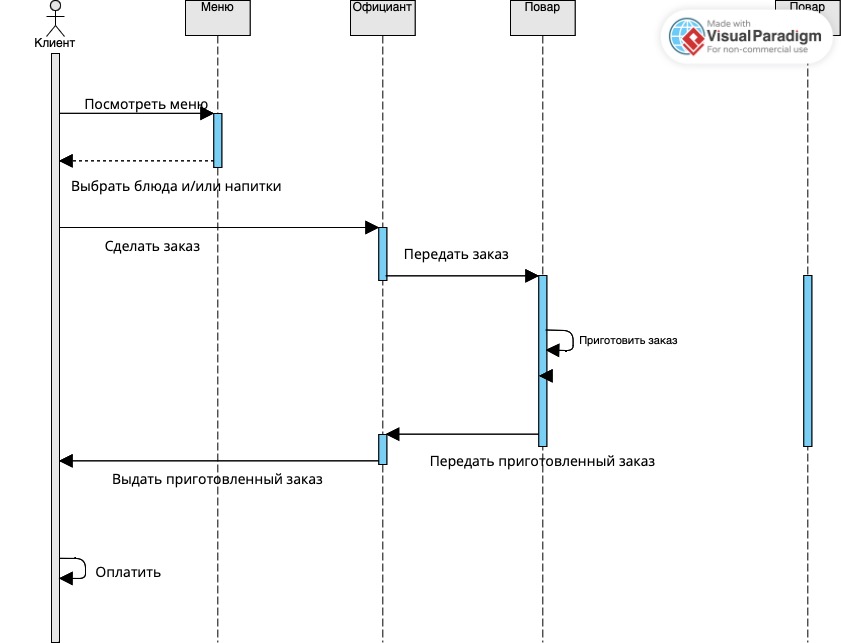


Рисунок 3 – Диаграмма последовательности

## 1.7 Диаграмма коммуникации

Диаграмма коммуникации, представлена на рисунке 3 и содержит объекты и сообщения между ними.

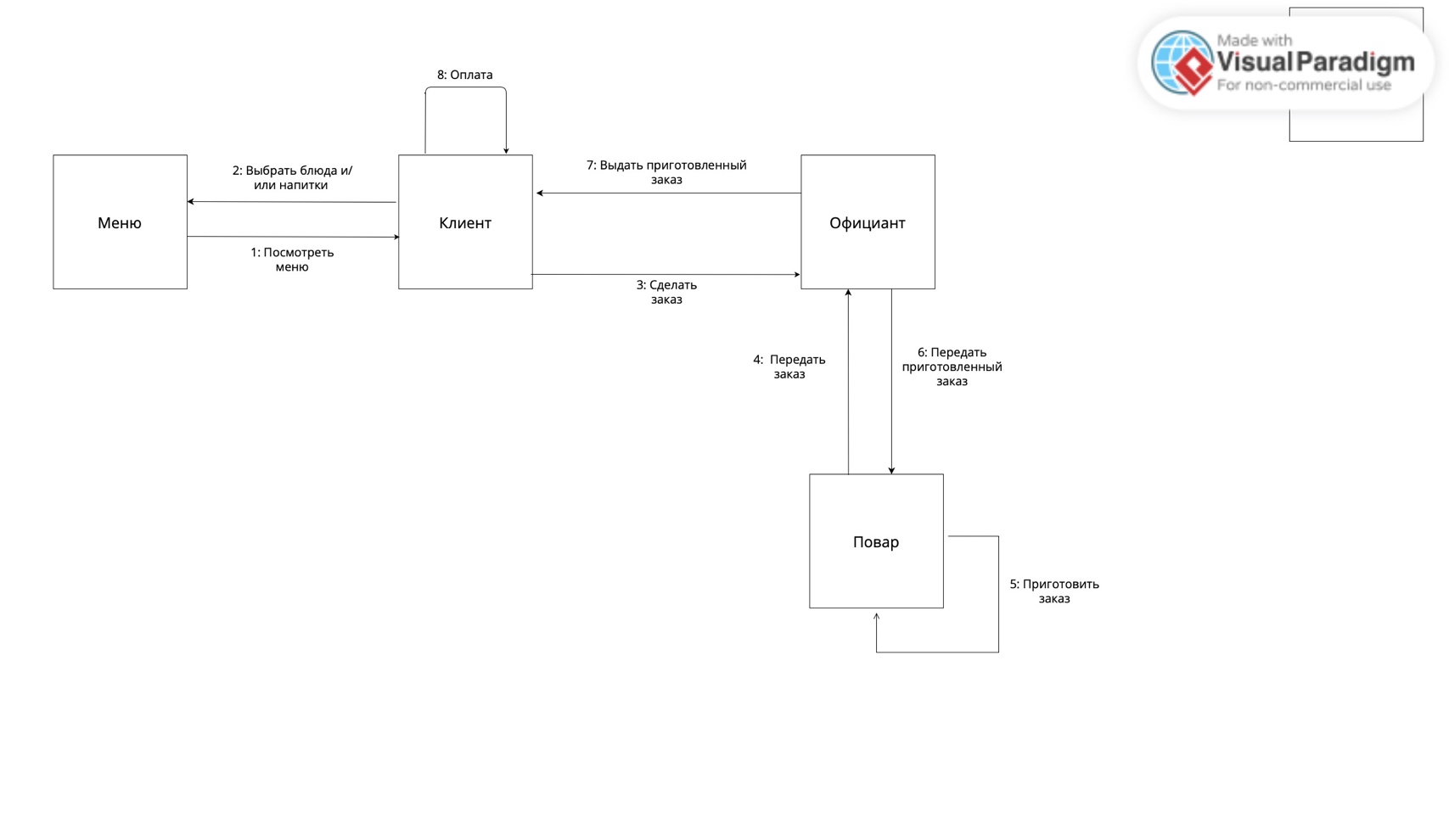


Рисунок 4 – Диаграмма коммуникации

## 1.8 Диаграмма деятельности

Диаграммы деятельности построены в зависимости от того как может быть использована программа.

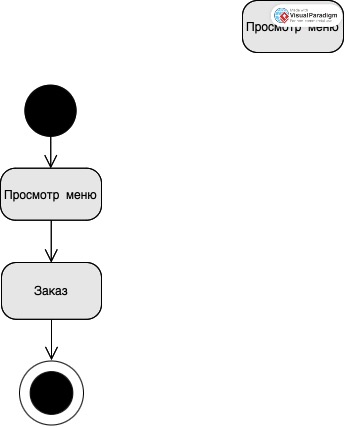


Рисунок 5 – Диаграмма деятельности «Клиент»

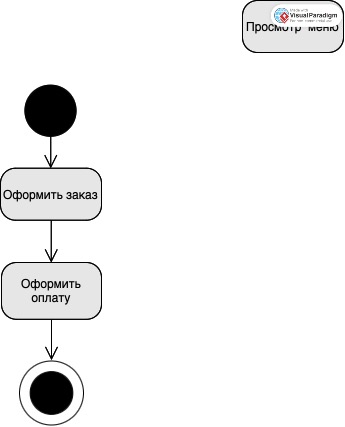


Рисунок 6 – Диаграмма деятельности «Официант»

# 2 Реализация

Проект реализован на языке программирования С#.

## 2.1 Описание классов

Программа состоит из нескольких классов:

Internal class Client – класс клиента

Поля:

- public int orderNumber – поле номера заказа

Методы:

- public void makeOrder() {} – метод совершения заказа

Код для класса отображен на рисунке 7.

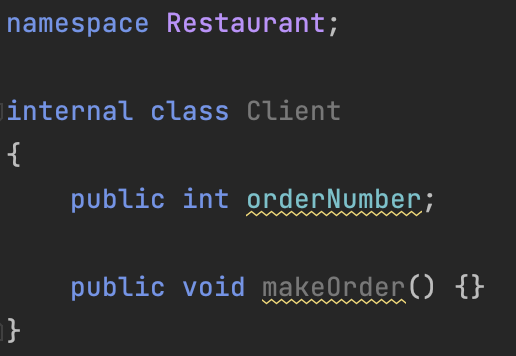


Рисунок 7 – Класс клиент

Internal class Menu – класс меню ресторана

Поля:

public string [] drinks – массив напитков в меню

public string [] meals – массив блюд в меню

Методы:

Public void printMenu() {} – метод вывода меню

Код для класса отображен на рисунке 8.

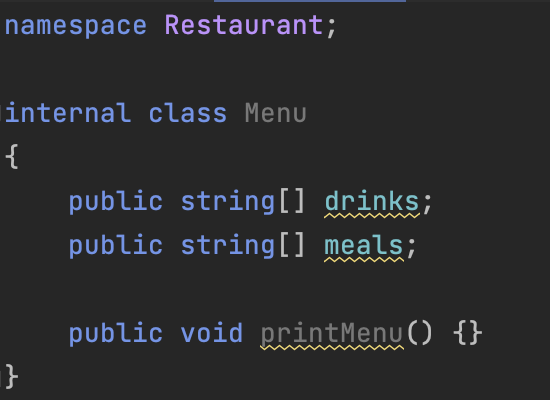


Рисунок 8 – Класс меню ресторана

Internal class Order – класс заказа клиента

Поля:

public int orderNumber – номер заказа

public int orderStatus – статус заказа

public bool payment – булеановое значение статуса заказа

Код для класса отображен на рисунке 9.

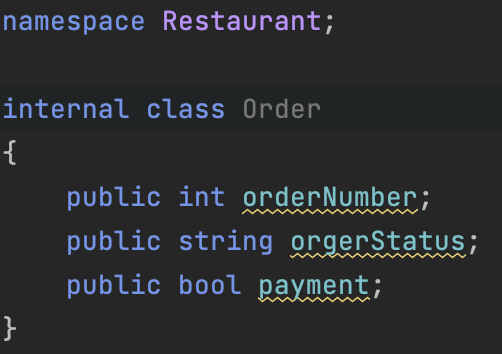


Рисунок 9 – Класс заказа клиента

Internal class Employees – родительский класс сотрудников ресторана

Поля:

public string Name – ФИО сотрудника ресторана

Код для класса отображ ен на рисунке 10.

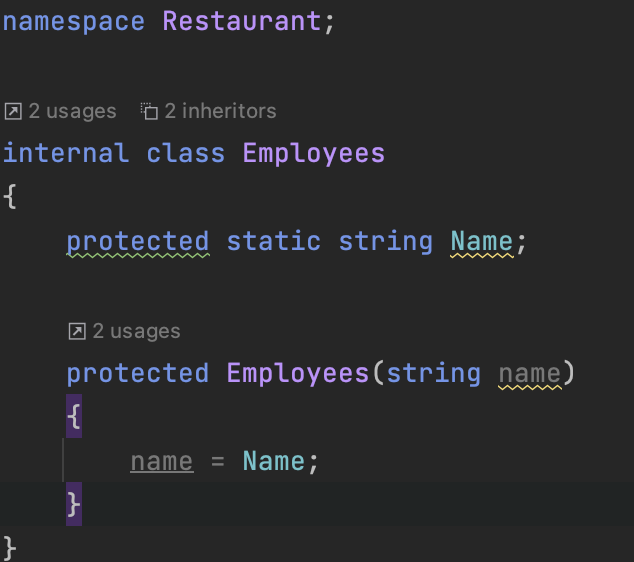


Рисунок 10 – Материнский класс сотрудников ресторана

Internal class Waiter – класс официанта

Поля:

string name – наследуемое ФИО из родительского класса Employees

Методы:

Public void formOrder() {} – метод формирования заказа

Public void makePayment() {} – метод оплаты заказа

Public void giveOrder() {} – метод выдачи заказа

Код для класса отображен на рисунке 11.

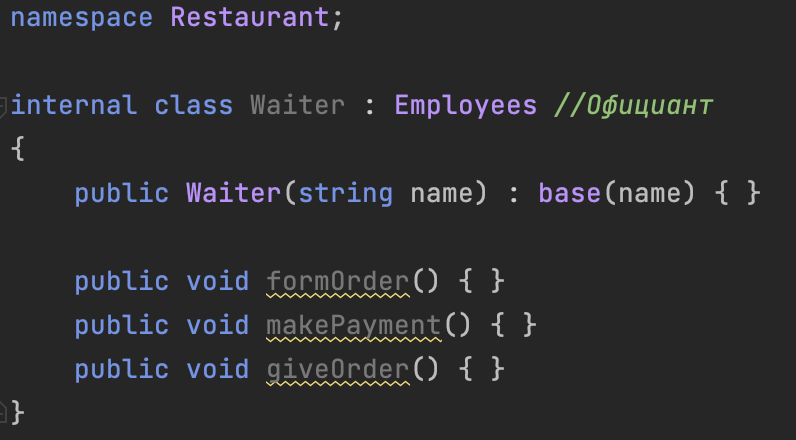


Рисунок 11 – Класс класс официанта

Internal class Chef – класс повара

Поля:

string name – наследуемое ФИО из родительского класса Employees

Методы:

Public void doOrder() {} – метод выполнения заказа

Код для класса отображен на рисунке 12.

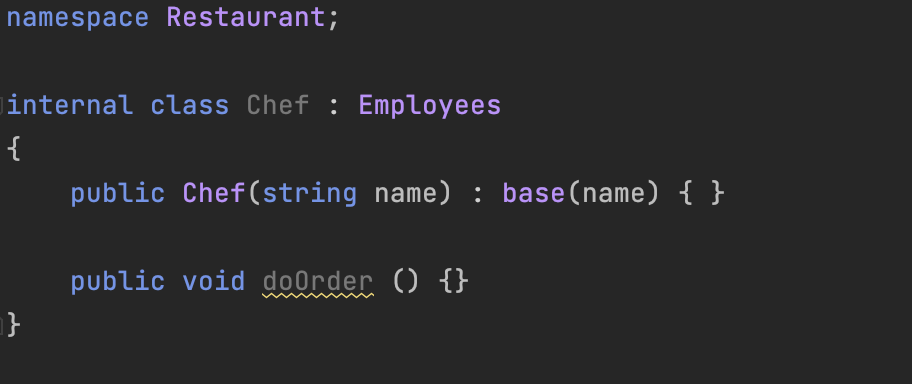


Рисунок 12 – Класс повара

# 3 Организация структуры папок проекта

В проекте находятся папки необходимые для описания классов.

Класс Client описывает клиента и содержит поле номера заказа и метод совершения заказа. Класс Menu описывает меню ресторана и содержит поля массивы напитков и блюд ресторана, а также метод вывода меню. Класс Order описывает заказ и содержит поля номера и статуса заказа, а также булеановое значение статуса заказа. Класс Employees является родительским классом сотрудников ресторана и содержит поле с ФИО сотрудника ресторана. Класс Waiter описывает официанта и содержит поле наследуемого ФИО из родительского класса Employees и методы формирования, оплаты, выдачи заказа. Класс Chef содержит поле наследуемого ФИО из родительского класса Employees и метод выполнения заказа.

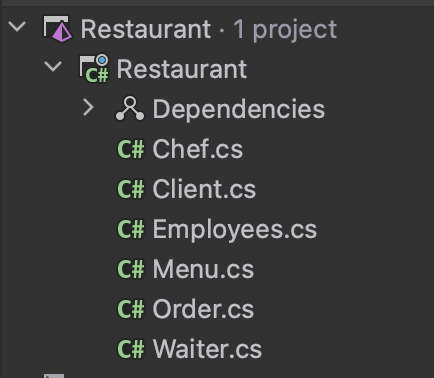


Рисунок 13 – Организация структуры папок

# Вывод

В ходе выполнения практического задания освоено изучение предметной области и реализация на языке программирования C#.