**Отчет по лабораторному занятию №9**

**Цель:**

Целью данной лабораторной работы было разработать многопоточное клиент/серверное приложение для заказа еды по сети с использованием TCP-соединения. Приложение должно обеспечивать взаимодействие клиента и сервера, где сервер предоставляет клиентам меню, а клиенты формируют заказ и отправляют его на сервер с указанием адреса доставки. Основная задача заключалась в реализации передачи данных между клиентом и сервером с использованием объектов Java, а также обеспечения многопоточности для обработки нескольких клиентов одновременно.

**Практическая часть:**

1. Описание архитектуры приложения

Приложение состоит из следующих основных компонентов:

* Класс Dish: представляет блюдо, содержащее название и цену.
* Класс Menu: хранит список блюд и предоставляет методы для добавления блюда и получения списка всех блюд.
* Класс Order: представляет заказ, включающий список выбранных клиентом блюд и адрес доставки.
* Класс FoodOrderClient: реализует клиента, который подключается к серверу, получает меню, формирует заказ и отправляет его серверу.
* Класс FoodOrderServer: реализует сервер, который предоставляет клиентам меню и обрабатывает поступающие заказы. Обрабатывает каждого клиента в отдельном потоке.
* Класс ClientHandler: вспомогательный класс, обрабатывающий подключение одного клиента в отдельном потоке.

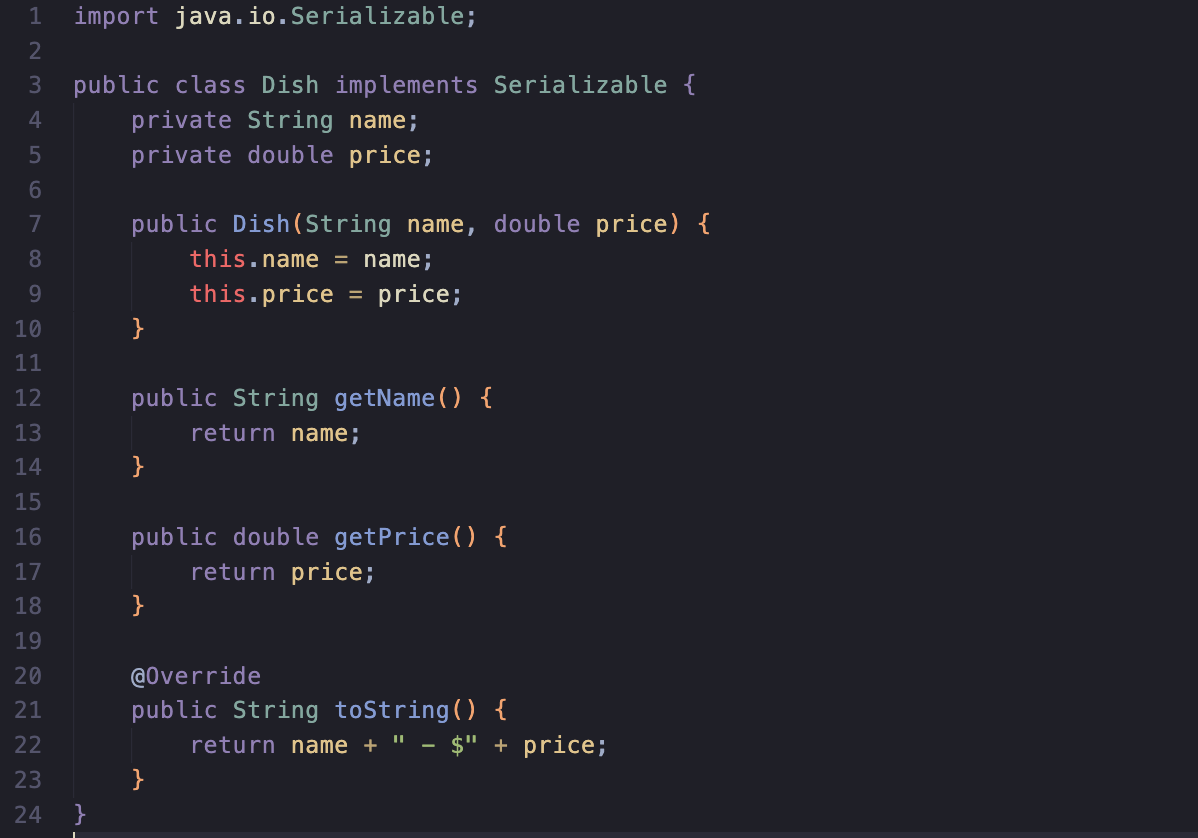
1. Реализация компонентов

Класс Dish реализует интерфейс Serializable, что позволяет передавать его объекты через сеть. Содержит поля:

* name — название блюда.
* price — цена блюда.

Методы:

* Конструктор для инициализации полей.
* Геттеры для получения значений полей.
* Метод toString() для строкового представления объекта.



Класс Menu также реализует интерфейс Serializable и содержит список блюд (List<Dish>). Предоставляет методы:

* addDish() — добавление блюда в меню.
* getDishes() — получение списка блюд.
* Переопределенный метод toString() для удобного отображения меню.
* Класс Order
* Класс Order хранит информацию о заказе:
* Список блюд, выбранных клиентом.
* Адрес доставки.

Класс также реализует Serializable, что позволяет отправлять его объекты через сеть. Метод toString()используется для вывода информации о заказе.

Класс FoodOrderClient

Клиентская часть приложения выполняет следующие задачи:

1. Подключается к серверу с использованием TCP-соединения.
2. Получает меню от сервера и отображает его пользователю.
3. Пользователь выбирает блюда из меню и формирует заказ.
4. Клиент запрашивает у пользователя адрес доставки.
5. Клиент отправляет сформированный заказ серверу.
6. Ожидает подтверждение от сервера и выводит его в консоль.

Используются ObjectOutputStream и ObjectInputStream для отправки и получения объектов через сеть.

Класс FoodOrderServer

Серверная часть приложения выполняет следующие задачи:

1. Инициализирует меню с несколькими блюдами.
2. Запускает серверный сокет и ожидает подключения клиентов.
3. При подключении клиента создает новый поток ClientHandler для обработки запроса.
4. Передает меню клиенту и получает от него заказ.
5. Выводит полученный заказ в консоль и отправляет подтверждение клиенту.

Сервер поддерживает многопоточность за счет использования класса ClientHandler, что позволяет обрабатывать нескольких клиентов одновременно.

Класс ClientHandler

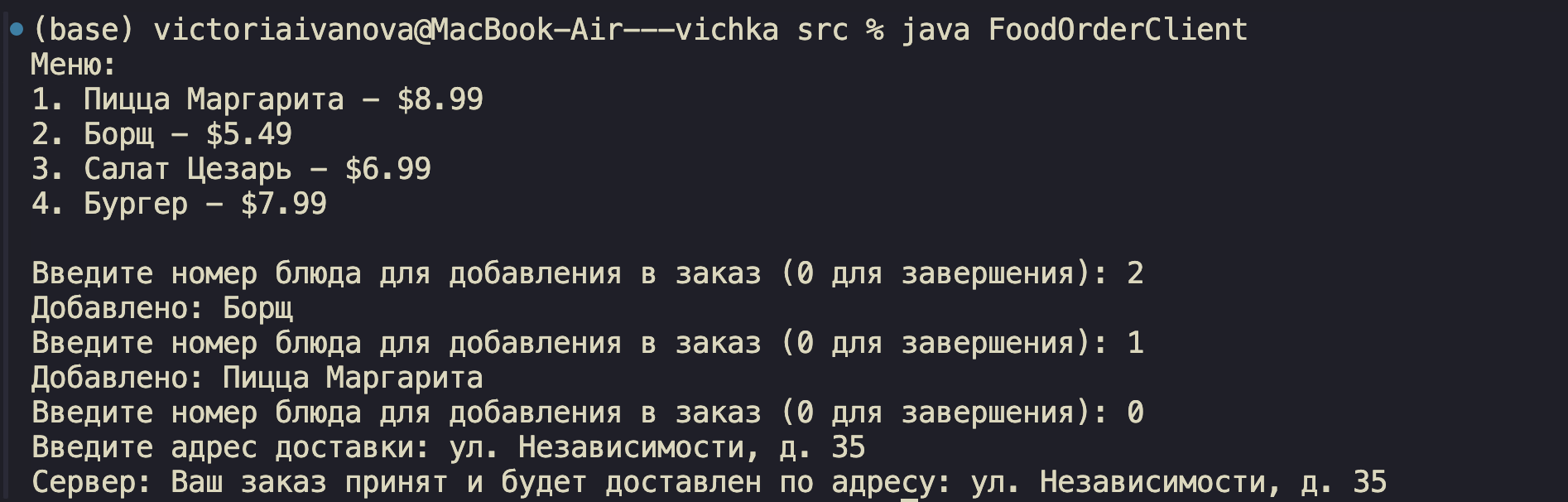
Этот класс является потоком, который обрабатывает одного клиента. В нем реализованы следующие этапы взаимодействия:

1. Отправка клиенту меню.
2. Чтение объекта заказа от клиента.
3. Вывод заказа в консоль сервера.
4. Отправка клиенту подтверждения получения заказа.

Протокол взаимодействия

Протокол взаимодействия основан на передаче Java-объектов через TCP-соединение с использованием ObjectOutputStream и ObjectInputStream:

1. Клиент подключается к серверу.
2. Сервер отправляет объект Menu клиенту.
3. Клиент отправляет объект Order серверу.
4. Сервер отправляет строку подтверждения клиенту.

**Пример работы:**

**Вывод:**

В результате выполнения данной лабораторной работы был разработан многопоточный клиент/сервер на основе TCP-соединения для оформления заказов на доставку еды. Приложение предоставляет пользователю удобный интерфейс для выбора блюд и формирования заказа, а сервер обрабатывает заказы клиентов и отправляет подтверждения.

Достигнуты следующие цели:

* Реализовано клиент/серверное взаимодействие с использованием Java TCP-сокетов.
* Использованы Java-объекты для передачи данных между клиентом и сервером.
* Обеспечена многопоточность на серверной стороне для одновременной работы с несколькими клиентами.