README.md для системы заказов еды

Этот код реализует упрощенную систему онлайн-заказа еды с использованием классов , , и на языке Python.MenuRestaurantClientOrder

Описание классов:

* **Menu:** Представляет блюдо в меню ресторана. Хранит категорию, название, цену, вес и ингредиенты блюда. Методы обеспечивает строковое представление блюда, добавляет блюдо в меню, выводит информацию о блюде, включая ингредиенты.\_\_str\_\_add\_menuprint\_menu
* **Restaurant:** Представляет ресторан. Хранит название ресторана, меню и данные о клиенте. Методы имитирует процесс оплаты заказа, отображает сообщение о принятии заказа.payment\_orderaccept\_order
* **Client:** Представляет клиента. Хранит имя, адрес и список заказов клиента. Методы запрашивает у пользователя имя и адрес, позволяет клиенту вводить заказы.input\_datainput\_order
* **Order:** Представляет заказ клиента. Хранит имя клиента, адрес доставки, список блюд, и статус доставки. Методы обеспечивает строковое представление заказа, вычисляет общую сумму заказа, выводит детали заказа, имитирует процесс доставки заказа.\_\_str\_\_sum\_orderprint\_orderdelivery\_order

Функциональность:

Система позволяет:

1. Создавать объекты для представления блюд в меню ресторана.Menu
2. Создавать объект , передавая в него меню и данные о клиенте.Restaurant
3. Создавать объект , получать от него данные (имя и адрес).Client
4. Принимать заказы от клиента и добавлять их в объект .Order
5. Выводить информацию о заказе, включая детали заказа и общую сумму.
6. Имитировать процесс оплаты и доставки заказа.

Использование:

Для использования кода необходимо создать экземпляры классов, заполнить меню и получить данные о клиенте. После этого можно оформить заказ, вывести его детали и имитировать процессы оплаты и доставки. Пример использования можно добавить в сам код.

Ограничения:

* Код является упрощенной моделью и не включает в себя обработку ошибок, взаимодействие с базой данных или графический интерфейс пользователя.
* Имитация оплаты и доставки осуществляется с помощью функции , что не отражает реальных процессов.time.sleep

Дальнейшее развитие:

* Добавление обработки ошибок.
* Интеграция с базой данных для хранения данных о меню, клиентах и заказах.
* Создание графического интерфейса пользователя.
* Реализация реальных процессов оплаты и доставки с использованием внешних API.

Этот README предоставляет базовое описание кода и указывает направления для его дальнейшего развития. Более подробное описание функциональности каждого класса и метода можно добавить непосредственно в комментарии к коду.

# README

## Описание

Данный проект представляет собой простую симуляцию ресторана с возможностью заказа блюд, оплаты заказа и отслеживания доставки. Программа позволяет пользователям вводить свои данные, выбирать блюда из меню, а также следить за статусом своего заказа.

## Структура кода

Код состоит из нескольких классов, каждый из которых выполняет свою уникальную задачу:

### Классы

1. \*\*Menu\*\*:

- Хранит информацию о блюде, включая категорию, название, цену, вес и ингредиенты.

- Содержит методы для добавления блюда в меню и печати информации о блюде, включая состав.

2. \*\*Restaurant\*\*:

- Представляет ресторан с названием, меню и клиентом.

- Содержит методы для обработки оплаты заказа и принятия заказа клиента.

3. \*\*Client\*\*:

- Хранит информацию о клиенте, включая имя, адрес и список заказанных блюд.

- Содержит методы для ввода данных клиента и выбора блюд из меню.

4. \*\*Order\*\*:

- Хранит информацию о заказе, включая данные клиента и список заказанных блюд.

- Содержит методы для вычисления суммы заказа, печати информации о заказе и отслеживания статуса доставки.

## Установка

Чтобы запустить программу, скопируйте весь код в файл с расширением `.py` (например, `restaurant\_simulation.py`). После этого запустите его с помощью Python:

```bash

python restaurant\_simulation.py

```

## Использование

1. Запустите скрипт.

2. Введите ваше имя и адрес доставки, когда будет предложено.

3. Выберите блюда из меню, вводя их названия. Чтобы завершить выбор, введите точку (`.`).

4. После завершения, программа предоставит информацию о вашем заказе, его статусе и итоговой сумме.

## Пример работы

```plaintext

Введите своё имя: Иван

Укажите адрес доставки: ул. Пушкина, д. 10

Клиент Иван введите блюдо из меню: Салат

Клиент Иван введите блюдо из меню: Паста

Клиент Иван введите блюдо из меню: .

Заказ клиента Иван:

Салат - 250 руб.

Паста - 400 руб.

Итоговая сумма заказа: 650 руб.

Статус доставки заказа клиента Иван по адресу ул. Пушкина, д. 10:

Собран

В пути

Доставлен

```

## Зависимости

Для работы программы используются стандартные библиотеки Python, поэтому дополнительных установок не требуется.

## Лицензия

Данный проект распространяется на условиях MIT License.

## Поддержка

Если у вас есть вопросы или вы столкнулись с проблемами, пожалуйста, создайте новый issue в репозитории или свяжитесь с автором.

---

Этот README файл предоставляет общее представление о проекте, его структуре и использовании. Вы можете изменять и дополнять его в зависимости от развития проекта и добавления новых функций.

**README ДЛЯ**

**«СИСТЕМА ОНЛАЙН-ЗАКАЗА ЕДЫ»**

**Описание**

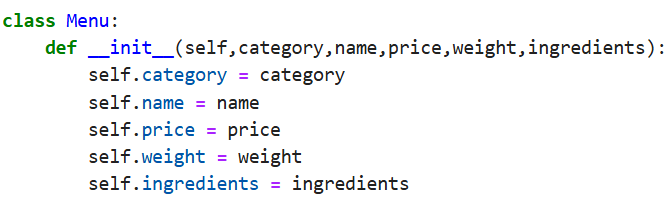
Данный проект представляет собой приложение, в котором пользователи могут просматривать меню, размещать заказы и отслеживать статус доставки.

Код программы позволяет вводить данные пользователя, вводить пользователям блюда из меню и отслеживать статус своего заказа. Код состоит из следующих классов, в которых есть необходимые методы, которые выполняют определенный задачи:

**Классы и методы**

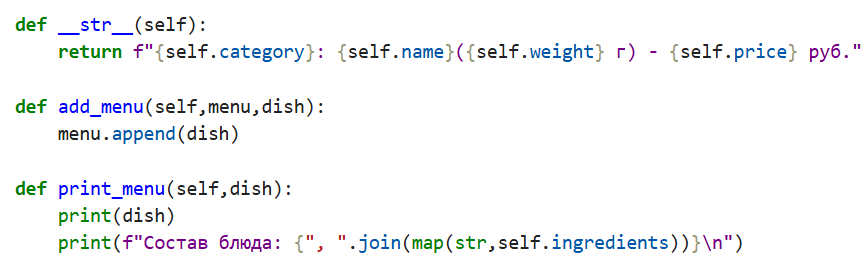
1. **Класс Menu**

В этом классе хранится информация о блюде: категория блюда, название блюда, цена, вес и состав блюда:

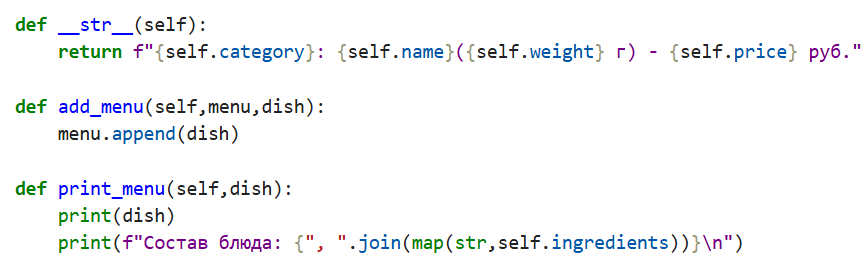


**Методы класса Menu:**

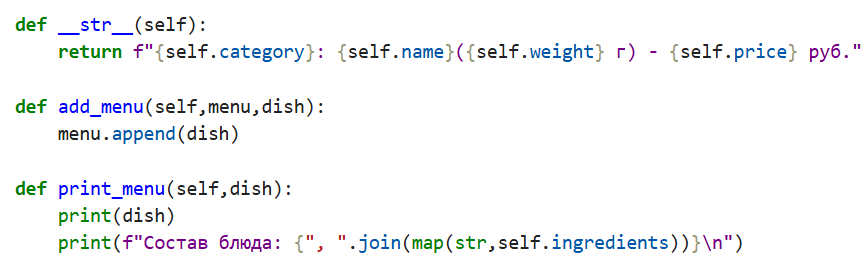
* Метод **\_\_str\_\_(self)** позволяет получить строковое представление об информации блюда, включая его название:



* Метод **add\_menu(self, menu, dish)** добавляет блюдо в список menu[]:

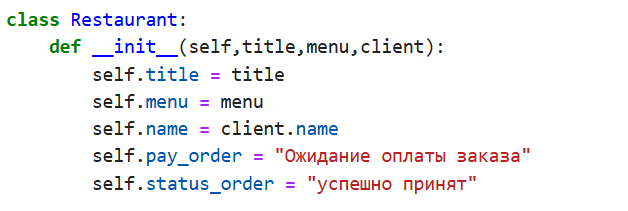


* Метод **print\_menu(self, dish)** выводит информацию о каждом блюде:



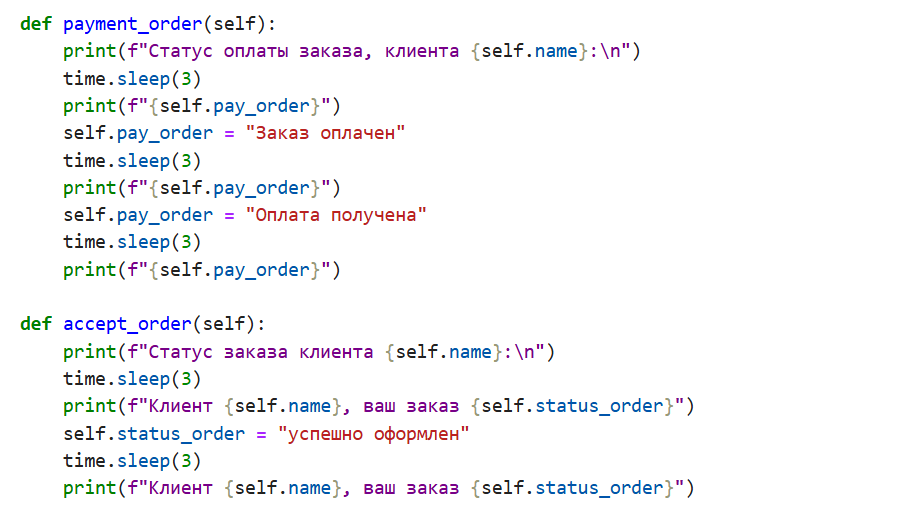
1. **Класс Restaurant**

Представляет ресторан с названием, меню и клиентом. Аргумент «клиент» является экземпляром **класса Client**, т.к. в классе Restaurant используется атрибут **класса Client** – «self.name», а также атрибуты «self.pay\_order» и «self.status\_order» для отображения статуса оплаты заказа и статуса заказа:

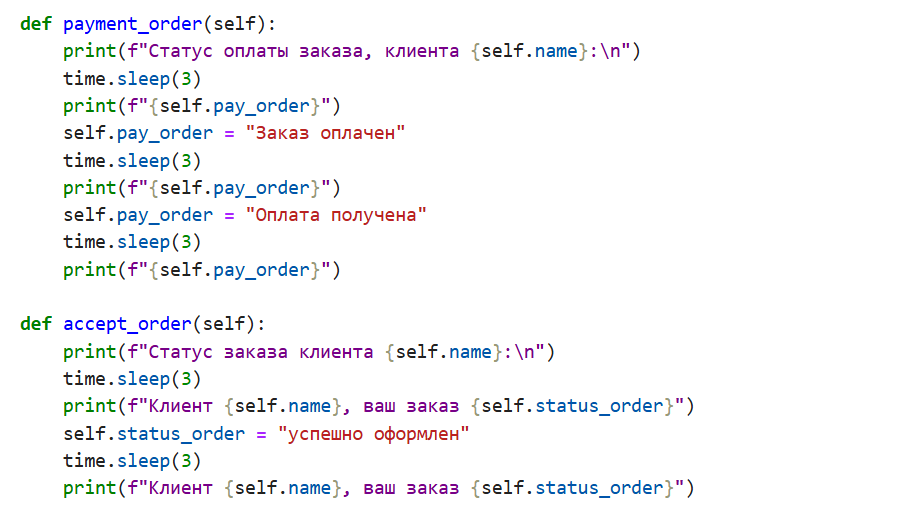


**Методы класса Restaurant:**

* Метод **pyment\_order(self)** выводит информацию о статусе оплаты заказа клиента, используя функцию **sleep** из модуля **time** для имитации отслеживания статуса оплаты:

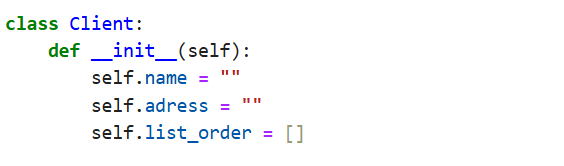
****

* Метод **accept\_order(self)** выводит информацию о статусе заказа клиента, используя функцию **sleep** из модуля **time** для имитации отслеживания статуса заказа:

****

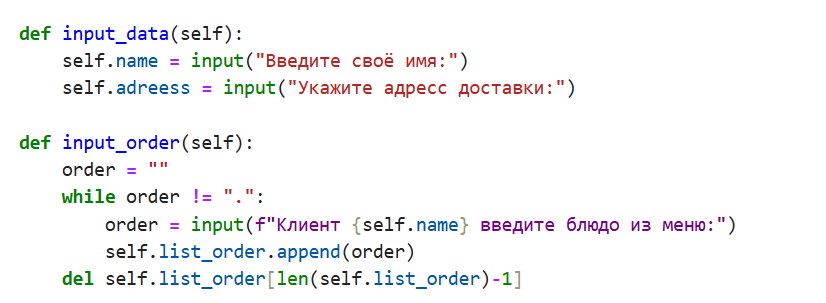
1. **Класс Client**

Содержит информацию о клиенте: имя, адрес доставки, а также список заказанных блюд клиентом:

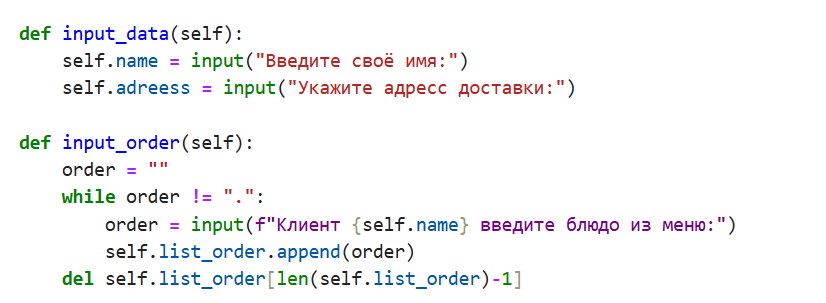


**Методы класса Client:**

* Метод **input\_data(self)** запрашивает у пользователя имя и адрес доставки:

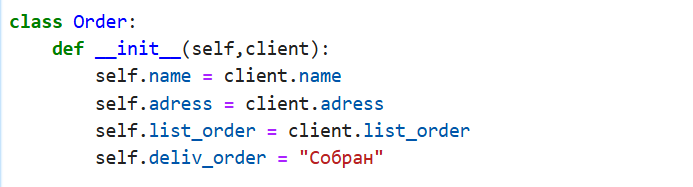


* Метод **input\_order(self)** просит ввести пользователя блюда из меню и добавляет каждое блюдо в список:



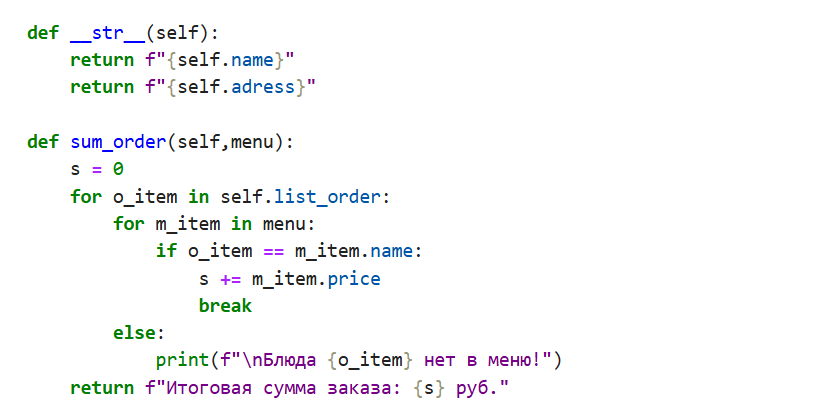
1. **Класс Order**

Хранит информацию о клиенте, заказе клиента и статусе доставки заказа:

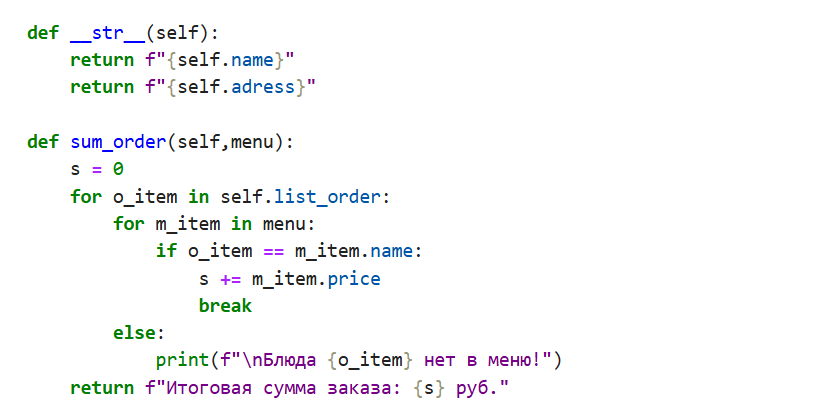


**Методы класса Order:**

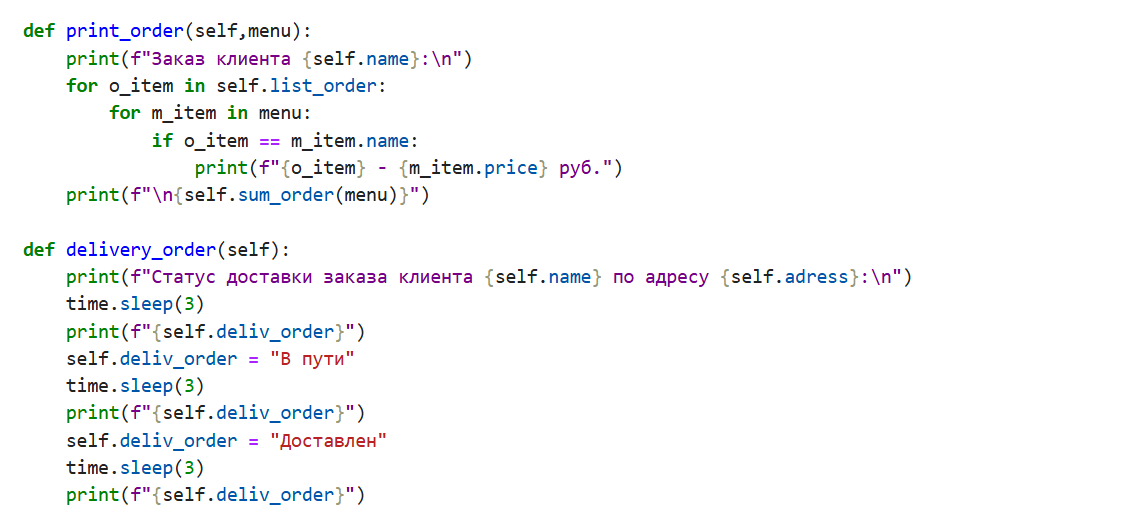
* Метод **\_\_str\_\_(self)** позволяет получить строковое представление об имени клиента и адреса доставки:

****

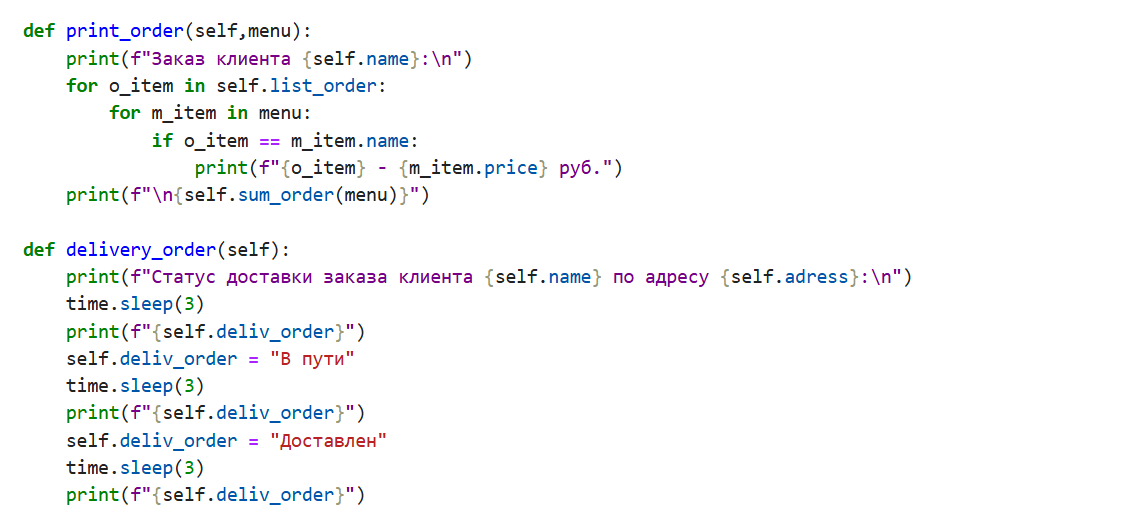
* Метод **sum\_order(self, menu)** считает сумму заказа клиента и выводит информацию, что блюда нет в меню, если клиент ввел неправильное название блюда:

****

* Метод **print\_order(self, menu)** подробновыводит заказ клиента:



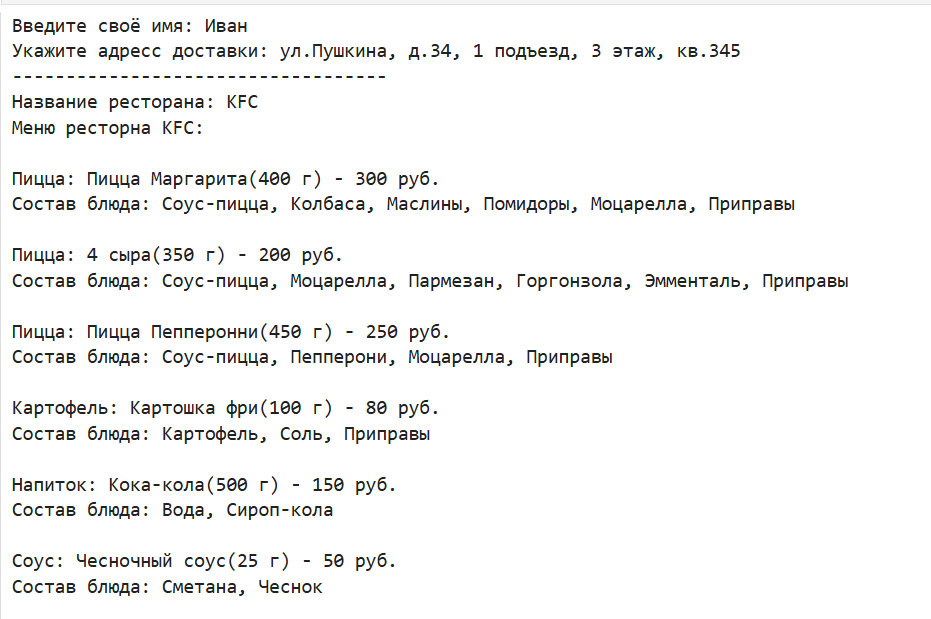
* Метод **delivery\_order(self)** выводит информацию о статусе доставки заказа клиента, используя функцию **sleep** из модуля **time** для имитации отслеживания статуса доставки заказа:

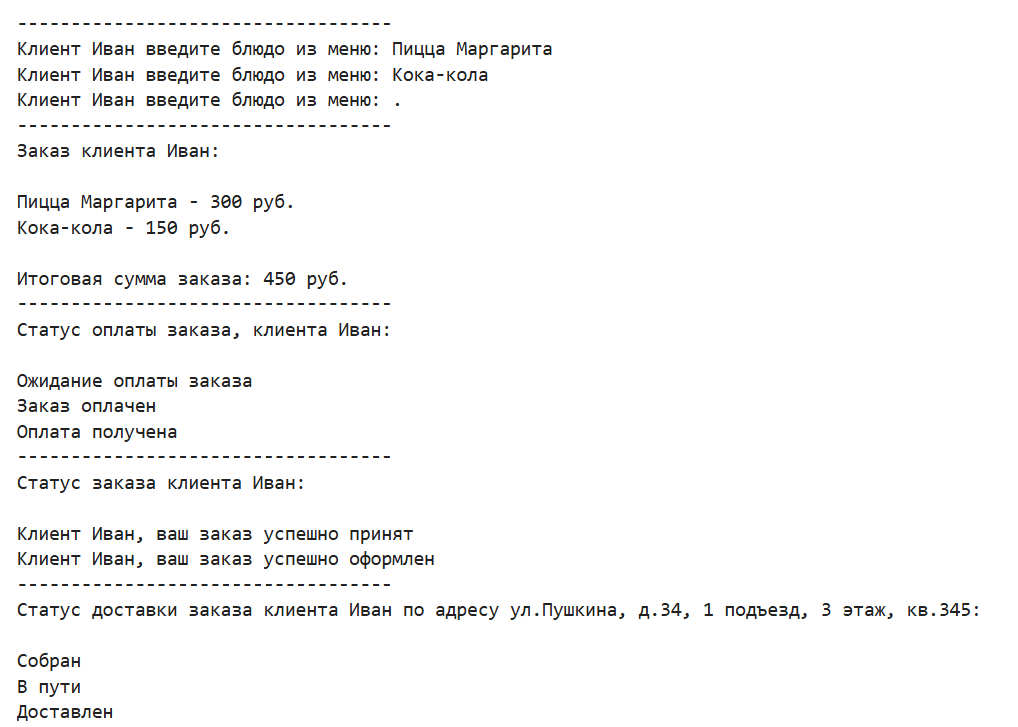


**Использование программы**

1. Запустите программу;
2. Введите ваше имя и адрес доставки, когда будет предложено.
3. Введите блюда из меню, вводя их названия. Чтобы завершить выбор, введите точку (**.**);
4. После ввода данных клиента и блюд из меню, программа выведет информацию о вашем заказе, итоговой сумме, а также статус заказа, статусы оплаты и доставки заказа.

**Пример работы программы**

****

****