**README ДЛЯ**

**«СИСТЕМА ОНЛАЙН-ЗАКАЗА ЕДЫ»**

**Описание**

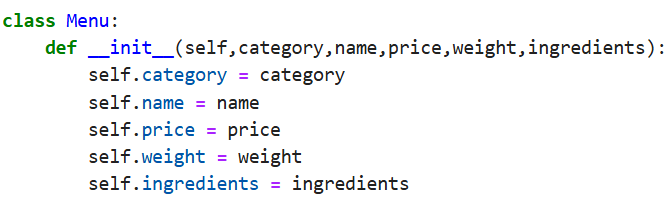
Данный проект представляет собой приложение, в котором пользователи могут просматривать меню, размещать заказы и отслеживать статус доставки.

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**

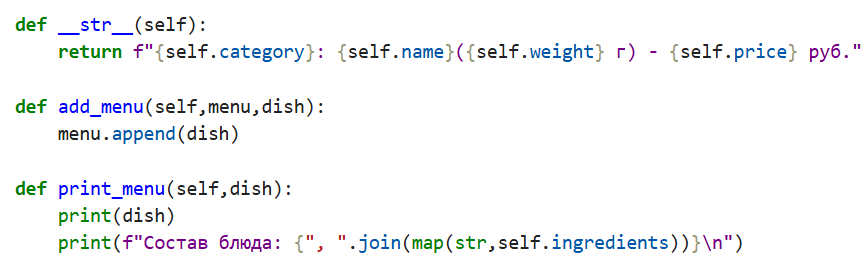
1. **Класс Menu**

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

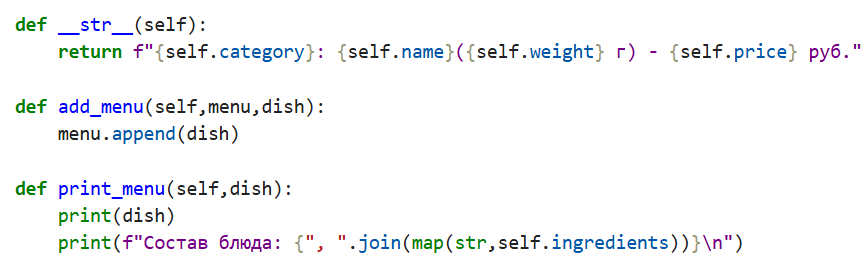


**Методы класса Menu:**

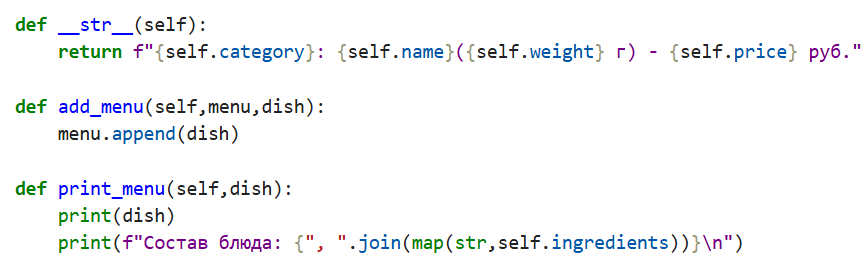
* Метод **\_\_str\_\_(self)** позволяет получить строковое представление об информации блюда, включая его название:



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

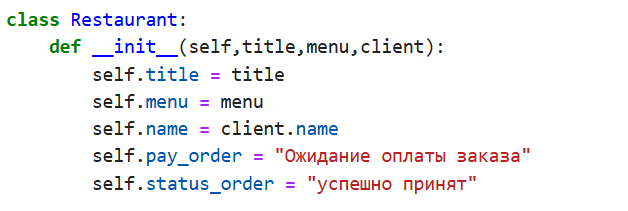


* Метод **print\_menu(self, dish)** выводит информацию о каждом блюде:



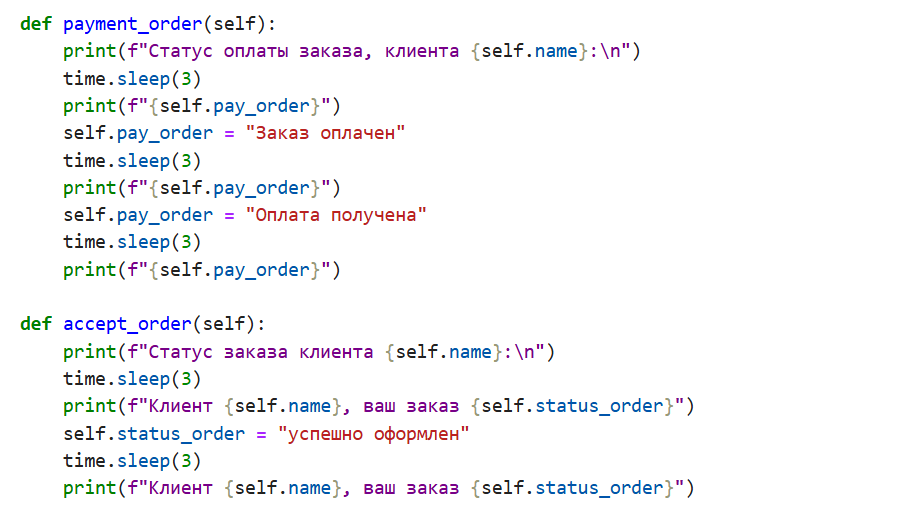
1. **Класс Restaurant**

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:

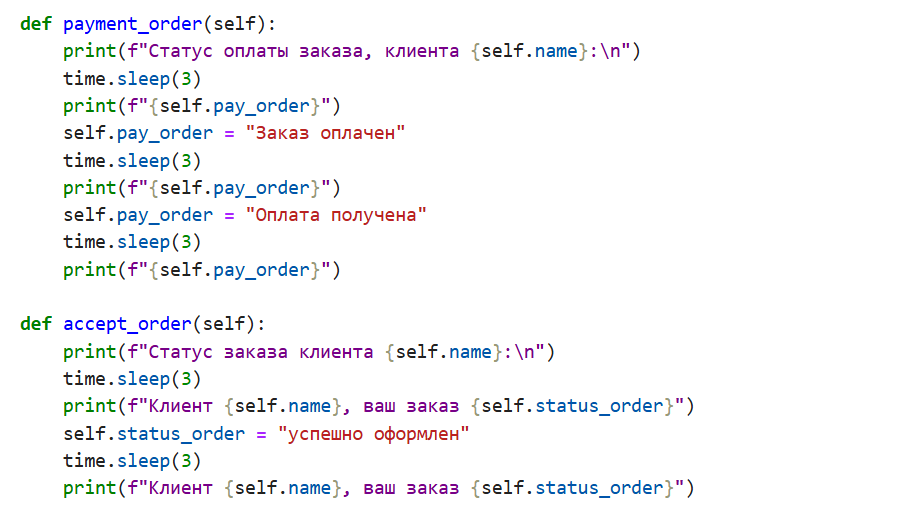


**Методы класса Restaurant:**

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

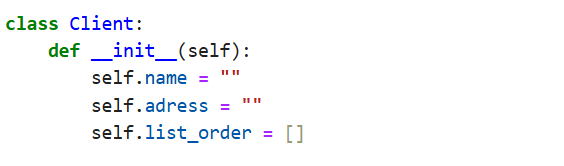
****

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

****

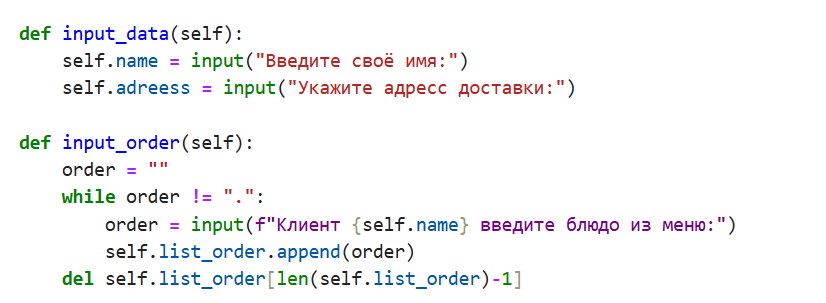
1. **Класс Client**

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

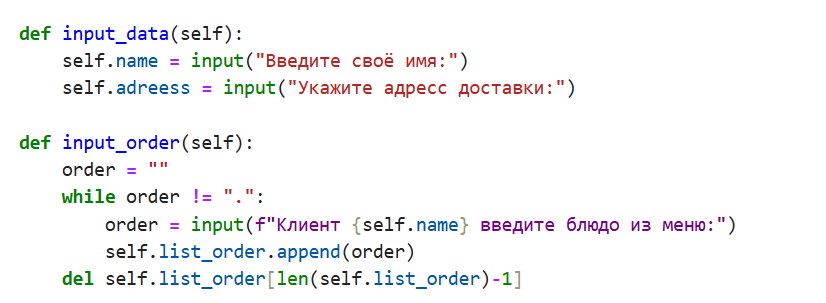


**Методы класса Client:**

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

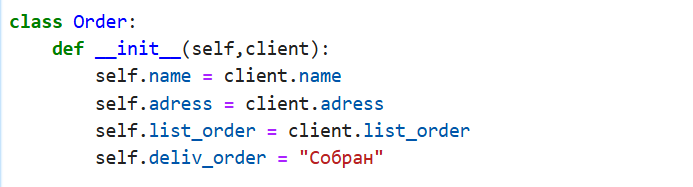


* Метод **input\_order(self)** просит ввести пользователя блюда из меню и добавляет каждое блюдо в список:



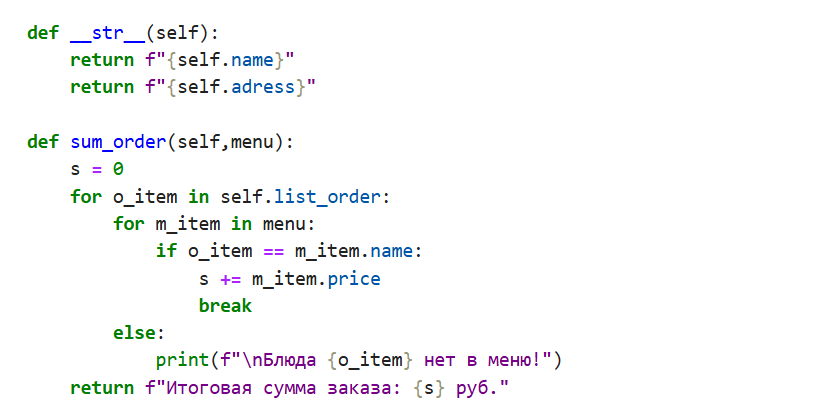
1. **Класс Order**

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

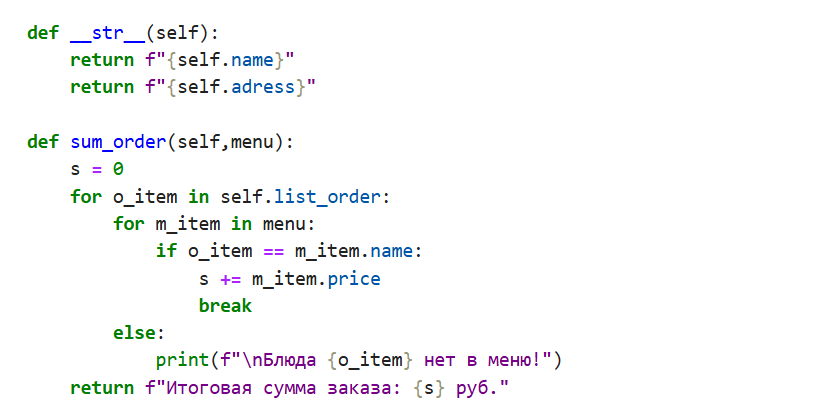


**Методы класса Order:**

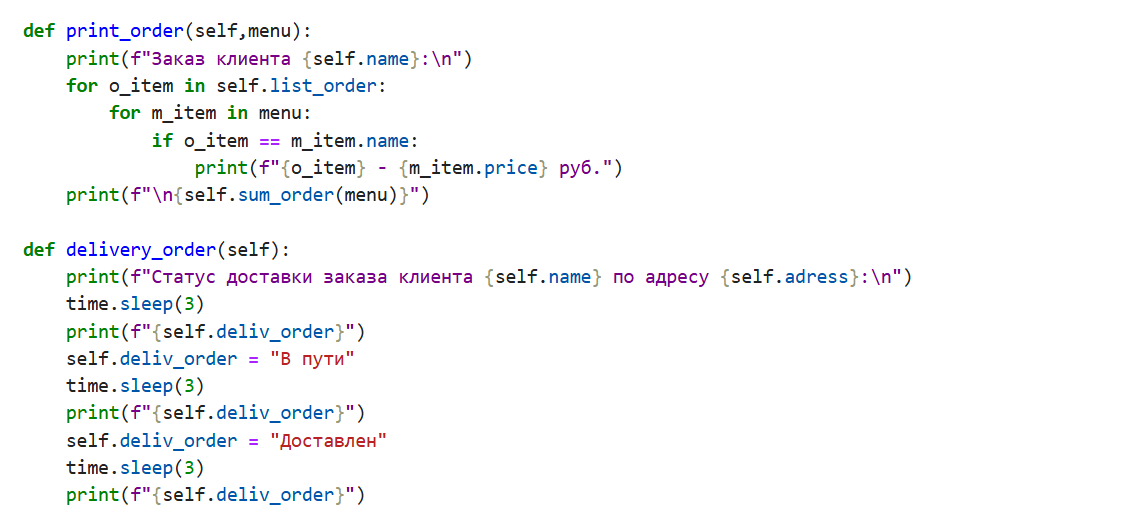
* Метод **\_\_str\_\_(self)** позволяет получить строковое представление об имени клиента и адреса доставки:

****

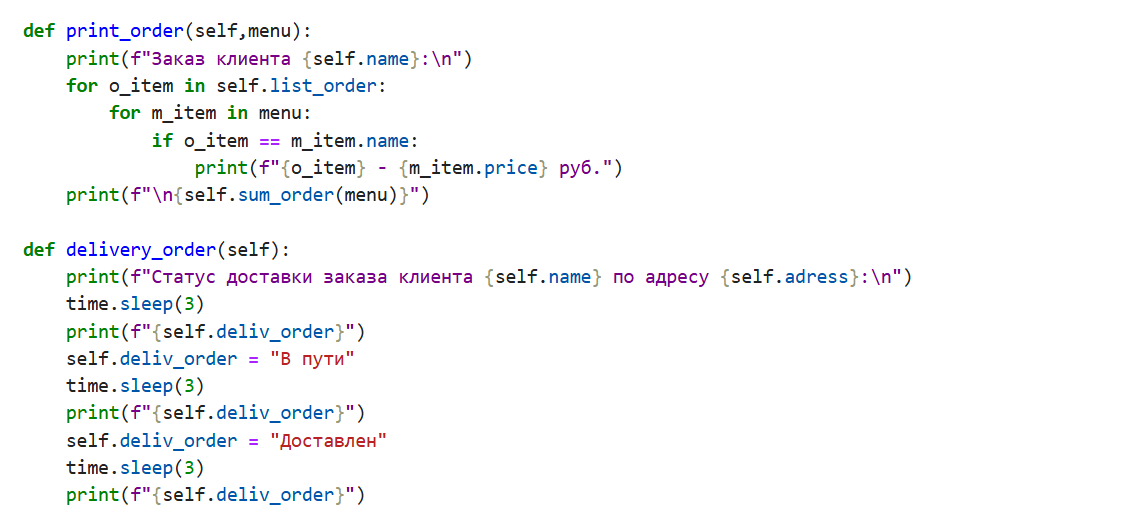
* Метод **sum\_order(self, menu)** считает сумму заказа клиента и выводит информацию, что блюда нет в меню, если клиент ввел неправильное название блюда:

****

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



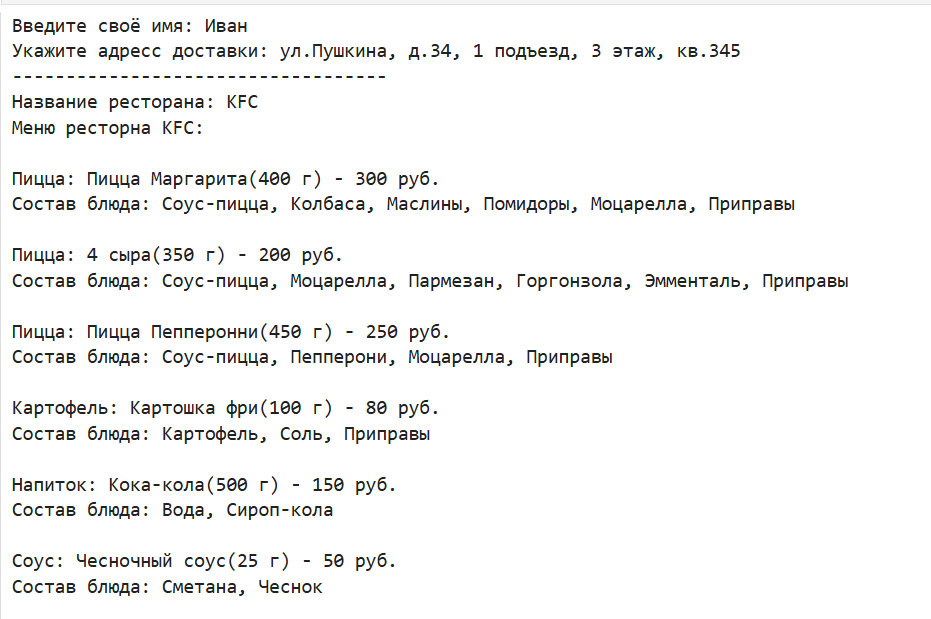
* Метод **delivery\_order(self)** выводит информацию о статусе доставки заказа клиента, используя функцию **sleep** из модуля **time** для имитации отслеживания статуса доставки заказа:

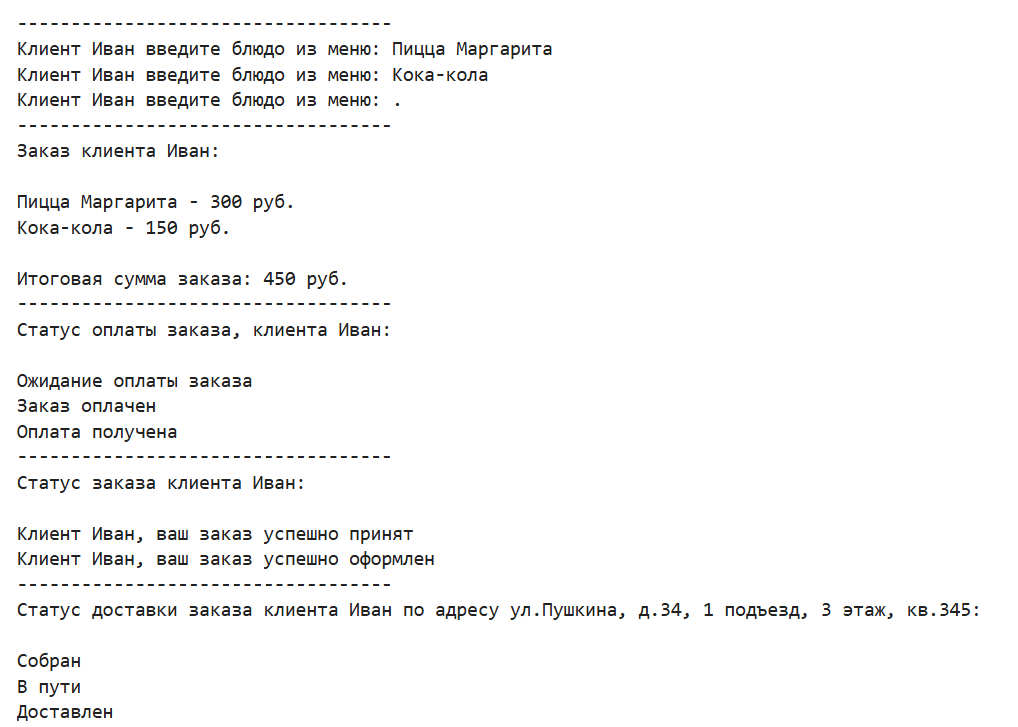


**Использование программы**

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**

****

****