

## Інтерфейси:

- IOffer
  - Властивості:
    - Ім'я
    - Опис
    - Категорія в меню (піца/десерт/напитки/гарячі страви/т.д.): Menu.Enum
    - Загальний час приготування: TimeOnly
- ICashRegister (Каса)//чи варто виділяти інтерфейс чи слід одразу класом
  - Властивості:
    - Замовлення
    - Перелік способів оплати
    - Кількість людей в черзі //уточнити чи потрібно
  - Методи:
    - Оплата (приймає ціну/замовлення і спосіб оплати)
- ISimulator
  - //надає доступ до дозволених ресурсів, наприклад візуалізатор спостерігатиме за подіями через нього. Можна організувати події.
- IVisio (придумати кращу назву)
  - //це інтерфейс візуалізатора із визначеними загальними методами для взаємодії з симулятором

//Вирішити чи потрібні два останні інтерфейси

## Класи:

- Chef
  - Поля:
    - Сховище для інгредієнтів: class
    - Сховище для результату роботи: class
    - Перелік/черга страв/елементів\_страв що має приготувати
    - Наступний кухар: Nullable<Chef>/замість кухара можна поставити інтерфейс для працівників кухні чи просто працівників
  - Властивості:
    - Що готує (категорія в меню): enum
  - Методи:
    - Додати страву у чергу -> void
    - Готувати -> void
    - Положити приготоване у відповідне місце -> void
  - Події:
    - Сповіщення про завершення (повідомляємо наступному кухарю після того як поклали приготоване у сховище)
- ChefManager:
  - Поля:
    - Перелік кухарів у розпорядженні
    - Нерозподілені страви для приготування: Dictionary<Order, Dictionary<IOffer, uint>>
  - Методи:
    - Встановлення для кожного кухара виду діяльності
    - Встановлення зав'язків між кухарами (хто кого повідомляє про завершення своєї частини)
    - Отримання нового замовлення та розподілення між кухарами

- OrderConstructor (Придумати кращу назву для працівника)
  - Поля:
    - Склад з якого бере приготоване кухарами
    - Список невиконаних замовлень
  - Методи:
    - Утворення замовлення із приготованих блюд
    - Передача замовлення (на касу чи деінде) / інформування про готове замовлення
- Order
  - Поля:
    - ID
    - Перелік елементів меню і кількість: Dictionary<IOffer, uint>
    - Загальна ціна
- Menu
  - Поля:
    - Перелік пропозицій з цінниками: Dictionary<IOffer, decimal>
    - Додатки до страв (на прикладі додатків до піци чи можливо інших страв): Dictionary< Enum, Dictionary<IOffer, decimal>> // тут IOffer можливо варто замінити
  - Enum:
    - Перелік категорій (піца/десерт/напитки/гарячі страви/т.д)
- Pizza : IOffer // за аналогією інші складні блюда (за їх наявності)
  - Enum:
    - Розміри
    - Тип основи
    - Етапи готовки //уточнити чи потрібно
  - Поля:
    - Розмір: enum
    - Тип основи: enum
    - Наповнення (чи варто замінити на словник із наповненням та кількістю): class
- Juice : IOffer // за аналогією кава/чай/інші напитки
  - Поля:
    - Об'єм
    - Смак
- Customer //переглядати віп-чи ні можна при оплаті. Порівнюється ідентифікатор покупця і наявність такого у базі. Як аналог надати покупцю поле із списком привілейованих рівнів
  - Поля:
    - Ідентифікатор (код/ПІБ/т.д.)
    - Поточний баланс коштів // мабуть це не є потрібним
    - Поточне замовлення що формує покупець
  - Методи:
    - Перегляд меню // чи зробити це властивістю (має меню – може переглядати/не має меню)
    - Формування замовлення -> Order
    - Здійснення замовлення через касу/термінал //чи варто цей і попередній пункт розділяти
- Manager
  - Поля:
    - Меню
    - Перелік акцій//щось типу Dict<VipLvl, List<IOffer, discount>> чи інших форматів
  - Методи:
    - Встановити меню

- Додати до меню
  - Вилучити з меню
  - Встановити акції
  - Додати до акцій
  - Вилучити з акцій
- Admin
  - Поля:
    - Менеджер/менеджери // чи варто віднести його/їх до працівників
    - Перелік працівників
  - Методи:
    - Видача інформації про працівника/менеджера
    - Додати вид Віп-клієнта
    - Вилучити вид Віп-клієнта
    - Додати/вилучити касу
    - Додати/вилучити спосіб оплати //можливо надати можливість налаштування каси
- User/SuperUser (чи якісь інші назви, або зробити його частиною адміна)
  - Поля:
    - Симулятор
  - Методи:
    - Встановлення швидкості симуляції//встановлення множника у скільки раз швидше ( $x > 1$ ) чи повільніше ( $x \in (0; 1)$ ) виконується симуляція
- CostumerGenerator
  - Поля:
    - Мінімальний час очікування
    - Максимальний час очікування
  - Методи:
    - Генерація покупця/покупців із випадковою затримкою в заданих межах//через yield чи як там реалізується безперебійна робота методу
    - Термінова генерація покупця/покупців
    - Термінова генерація заданої кількості покупців
- Simulator : ISimulator // до нього підключається візуалізатор і тяне дані для відображення

//Додаткові класи:

- Інформаційна дошка – на неї менеджер вішає меню, оголошує акції і з неї покупці можуть отримати інформацію про це
- Клас із енумом з назвами віпок
- Працівник для видачі замовлень. Чи його роботу виконуватиме пакувальник?
- Якщо надати кухарям білдери, то можна створити абстрактний клас білдер і від нього окремо під кожну страву новий білдер. Головний на кухні коли роздає кухарям ролі, може визначити хто яким білдером користуватиметься.