Pizza shop

Twoim zadaniem jest zaprojektowanie systemu przyjmowania zamówień, obsługi kuchni i wydawania zamówień w pizzerii. Aplikacja będzie zainstalowana na komputerze kucharza oraz dostawcy. Przyjmowanie zamówień powinno się odbywać w sposób zautomatyzowany. Założenia działania aplikacji:

1. Powinna umożliwiać przyjęcie zamówienia od klienta w postaci tekstu naturalnego możliwość zamówienia pizzy o nazwie (np. capriciossa), np. klient pisze:
    a. „Potrzebuję pizzę z serem, salami i oliwkami na Opolska 5” powinno skutkować przesłaniem do kuchni zamówienia zawierającego składniki: ser, salami i oliwki z adresem Opolska 5
    b. „Poproszę ser, salami na Katowicka 20” powinno skutkować przesłaniem do kuchni zamówienia zawierającego składniki: ser, salami z adresem Katowicka 20
    c. „Oliwki, cebulę na Bkaasdawe2 12” powinno skutkować przesłaniem do kuchni zamówienia zawierającego składniki: oliwki, cebula z adresem Bkaasdawe2 12
    d. „Xxxxas na Katowicka 202” powinno skutkować odrzuceniem zamówienia (odpowiedni komunikat dla użytkownika)
2. Zamówienie powinno być przesyłane do kuchni
3. W kuchni po odebraniu zamówienia przez kucharza (opcja gotuj) wypisać zamówienie, które jest pierwsze w kolejce do obsłużenia. Kucharz błyskawicznie je realizuje, oznacza pizze jako gotową i zdejmuje zamówienie z kolejki. Jakość upieczonej przez kucharza pizzy jest losowa ☹
4. Dostawca czeka zawsze na 5 pizz i dopiero wtedy wyrusza (automatycznie). Każda z pizz dodatkowo zawiera adres na który powinna zostać wysłana. Informacje o jej składnikach i adresie do dostawy powinny zostać wyświetlone.
    a. Następnym krokiem byłoby zastanowienie się czy nie bardziej efektywnie byłoby wysyłać dostawcę dopiero kiedy uzbiera się więcej pizz pod wskazany adres
5. Po zrealizowaniu dostawy dostawca dodaje wpis do danych na podstawie których może zostać wygenerowany raport.

6. Dane z których generowany jest raport przechowywane są w pliku dane.csv, w formacie:
   Adres;Składnik1,Składnik2,Składnik3;Jakość np.
   Opolska 5;CHEESE,SALAMI,OLIVES;BEST
   Katowicka 20;OLIVES,ONION;WORST
7. Powinno być możliwe wygenerowanie raportu (komenda „raport”), raport powinien wyświetlać informacje:
   a. Jaka ulica najczęściej była wskazywana do dostawy
   b. Jaki był najpopularniejszy składnik
   c. Jaka była średnia jakość pizzy (przyjmując, że BEST to 10, MEDIUM 5, a WORST 0)
8. *Dodaj nową jakość, powinna do tego wystarczyć zmiana jednego pliku, zastanów się co trzeba byłoby zmienić gdyby nie odpowiednie zastosowanie enuma (a więc co zyskaliśmy stosując enuma).
9. *Dodaj 2 dodatkowe składniki możliwe do dołożenia do pizzy
10. *Dodaj obsługę podawania konkretnych nazw pizz zamiast dyktowania konkretnych składników, np. Capriciosa ma grzyby, salami i ser, Margarita ser i oregano itd.
11. *Dodaj obsługę wyliczania cen na podstawie wybranych składników, powinna być wyświetlana klientowi przy zamówieniu

12. Instrukcja:

1. Obsługa przyjmowania zamówienia:
    a. W metodzie OrderService.takeOrder:
        i. za pomocą String.contains, String.indexOf i String.substring lub (nawet lepiej) wyrażeń regularnych sprawdź co zawiera zamówienie i przetwórz je na obiekt Order. Wszystkie dostępne składniki możesz uzyskać za pomocą Ingredient.values(). **UWAGA: jak wykrywać w miarę celnie wykrywać składniki dowiesz się na zajęciach, wskazówka - lematyzacja**
        Zakładamy, że zamówienie zawsze na końcu zawiera na <tutaj podany adres>, czyli jest skonstruowane według schematu: <Tekst, który powinien zawierać składniki> na <adres> (przykłady są w ogólnych założeniach)
        Po przetworzeniu słów klienta powinien zostać utworzony nowy obiekt Order z przypisanymi odpowiednimi składnikami i adresem, pole uuid zostaje wypełnione automatycznie (podejrzyj kod klasy Order). Nowo stworzony obiekt powinien zostać dodany do kolejki orders znajdującej się w OrderService. Jeśli tekst wprowadzony przez klienta nie zawiera żadnych składników wyrzuć wyjątek InvalidUserInputException.
2. Obsługa gotowania:
    a. W metodzie OrderService.viewTopCookingRequest zwróć kolejny element z kolejki przetworzony na obiekt klasy CookingRequest, powinno się tam znajdować uuid zamówienia i składniki pizzy. **UWAGA: element nie powinien zostać zdjęty z kolejki, a jedynie „podejrzany”.**
    b. W metodzie CookingService.cookNext() powinna być pobierana informacja o następnym zamówieniu w kolejce za pomocą metody OrderService.viewTopCookingRequest.
    Następnie na podstawie zwróconego z tej metody obiektu CookingRequest powinien zostać stworzony obiekt Pizza na podstawie składników z CookingRequest i o losowej jakości.

Na koniec powinna zostać wywołana metoda OrderService.markPizzaAsReady(pizza) z przed chwilą utworzonym obiektem typu Pizza w argumencie.

c. W metodzie OrderService.markPizzaAsReady powinien zostać **zdjęty** element z kolejki orders. Ten element i argument metody (Pizza) powinny posłużyć do utworzenia obiektu typu PizzaWithAddress. Nowo utworzony obiekt powinien zostać dodany do kolejki readyOrders.

3. Dostawa:
    a. W Runner po wpisaniu każdego polecenia (a może tylko po wpisaniu niektórych? – optymalizacja na później) powinna być sprawdzana ilość gotowych do rozwiezienia pizz za pomocą OrderService.readyPizzasCount(). Jeśli ich ilość jest równa 5, powinien nastąpić proces rozwożenia (powinna zostać wywołana metoda deliveryService.sendTheTruck() zawierająca w argumencie pizze z kolejki
    b. W DeliveryService.sendTheTruck wyświetl wszystkie wysłane pizze (składniki + adres), następnie wywołaj metodę ReportService.addRecord po kolei ze wszystkimi pizzami, które zostały wysłane do dostawy
4. Raportowanie:
    a. Metoda ReportService.addRecord powinna zapisywać w pliku dane.txt dane w formacie:
    Adres;Składnik1,Składnik2,Składnik3,Składnik…;Jakość
    np.
    Opolska 5;CHEESE,SALAMI,OLIVES;BEST
    Katowicka 20;OLIVES,ONION;WORST
    b. Metoda ReportService.printReport powinna odczytywać dane z pliku, parsować je (a więc zamieniać linijkę tekstu na obiekt typu PizzaWithAddress z wypełnionymi wszystkimi polami), po przeparsowaniu danych powinniśmy otrzymać List<PizzaWithAddress> z pizzami, które były w pliku. Na podstawie danych zawartych w liście należy wyznaczyć:
        i. Ile sumarycznie było zamówień
        ii. Jaka ulica najczęściej była wskazywana do dostawy

iii. Jaki był najpopularniejszy składnik
iv. Jaka była średnia jakość pizzy (przyjmując, że BEST to 10, MEDIUM 5, a WORST 0)