Модельовані структури даних Python. Черга

Задача. Побудувати програмну модель організації прикордонного митного контролю. Використати структуру даних черги.

Подібні задачі виникають в різноманітних сервісах, пов'язаних з організацією прикордонних служб.

Задачу моделювати для вантажних перевезень. Вантажні авто прибувають до кордону, стають в чергу на митний огляд, проходять огляд, після огляду рухаються далі до пункту призначення або отримують інші розпорядження.

Створити власний клас черги:

*1) з методами **базовими** для черги як структури;*

*2) з методами **зв'язаними** зі змістом задачі;*

*3) з методами **контролю і захисту** (питання безпеки і стійкості).*

Частина 1. Побудувати шаблон митної декларації. Митна декларація повинна мати всі необхідні відомості про перевезення товару. Шаблон будувати на основі структури словника. Для прикладу, шаблон відомостей може мати такі ключі і значення (на змістовому рівні):

```
avtonn = { авто : марка, номер_авто : номер, власник_авто : компанія,
           дата : рік-місяць-день,
           звідки : (країна,місто), куди : (країна,місто),
           товари : [ перелік_товарів – (назва,вартість) ],
           договір_перевезення : є / нема,
           інші_дані : . . . . .
         }
```

Будемо вважати для спрощення задачі, що кожне авто перевозить лише один вид товару. Точний зміст шаблона визначити самостійно.

Частина 2. Побудувати окремо список митних декларацій авто, які мають прибути до кордону (є в дорозі) і пройти митний конторолль. Список будувати в довільному порядку і використати умовні дані (але правдоподібні). Для будови списку можна використати шаблон і кожне авто заповнити вручну. Можна дещо автоматизувати, врахувавши, що шаблон завжди буде мати однаковий перелік ключів, а змінними будуть лише значення. Наприклад:

```
listkey = [ авто, номер_авто, власник_авто, дата, . . . ]
           # список містить перелік ключів шаблона
data1 = [ "Volvo", D1093AH, "Green day company", 2020.10.12,
          ("Poland","Wroclaw"), ("Україна","Вінниця"), . . . ]
avto1 = dict ( zip ( listkey, data1 ) ) # декларація для одного авто
avto2 = . . . . .
```

Частина 3. Будемо використовувати дві черги: загальна, зелений коридор. З кожною чергою можна виконати операції відповідно до функцій програмного класу черги.

Частина 4. Операції митного контролю. [*Методи,* ***зв'язані*** *зі змістом задачі*]

- поставити авто в чергу для огляду; (одну або другу)
- закінчити огляд і дозволити перетин кордону – читати з черги;
- перевести авто з однієї черги в кінець другої;
- викреслити авто з черги (немає дозволу на перетин кордону);
- скласти перелік авто (місце в черзі), які везуть вказаний товар;
- яке авто має товар найбільшої вартості?
- таблиця товарів і цін, перевезених через кордон (підсумок);
- які авто прямують до Одеси?
- інші операції – додати самостійно.

Частина 5. Диспетчер пункту прикордонного контролю. Скласти довільний сценарій комбінації перечислених операцій митного контролю (частини 4) і відповідну програму. Авто вибирати з переліку частини 2. Друкувати необхідні результати операцій в протоколі.

Сценарій має передбачати в тому числі спроби некоректих операцій. Відповідно має бути відображено в протоколі реагування методів контролю і захисту.

Вимоги до реалізації

1) Завдання реалізувати в цілому за технологією ООП. Весь програмний код оформити одним модулем (файлом).
2) Реалізувати власний клас черги, зовнішній, побудований з базовими і зв'язаними методами, доступний для всіх функцій. На його основі виконати програмування завдань.
3) Кожен метод частини 4 завдання коментувати за текстом програми, на забути пояснити параметри методів.
4) Побудувати і пояснити сценарій, визначений як частина 5 завдання. Показати в своєму сценарії приклади використання кожного метода частини 4 завдання. Надрукувати сценарій в окремий текстовий файл. Друкування результатів виконання має супроводжуватись коментарями у файлі чи на екрані.
5) Окремо надіслати скріншоти виконання на своєму комп'ютері, або протокол виконання записаний в єдиний текстовий файл.
6) Щоб проєкт мав належний рівень, список митних декларацій (частина 2) має бути в обсязі 15-20 елементів або більше.

---

7) Можна використати як допоміжний спеціальний клас даних модуля `dataclasses` (для частини 1 і частини 2 завдання):
https://realpython.com/python-data-classes/
https://docs.python.org/3.7/library/dataclasses.html?highlight=dataclasses#module-dataclasses

В результаті надіслати у відповідь такі файли:

- файл програмного коду;
- файл класу черги – якщо клас визначений окремо;
- файл з списком митних декларацій – якщо визначено окремож
- текстовий файл протокола виконання – обов'язково.

Файли надіслати окремими вкладеннями, НЕ АРХІВУВАТИ !