

## Модельовані структури даних Python. Черга

Задача. Побудувати програмну модель організації прикордонного митного контролю. Використати структуру даних черги, *створити власний клас з методами базовими для черги як структури, і зв'язаними зі змістом задачі – це найперша вимога до виконання завдання.*

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.

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 шаблону визначити самостійно.

Частина 2. Побудувати окремо список митних декларацій авто, які мають прибути до кордону (є в дорозі) і пройти митний контроль. 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 шаблону
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 – вимога* ]

- ❖ 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 перелічених операцій митного контролю (частина 4) і відповідну програму. Авто вибирати з переліку частини 2. Друкувати необхідні результати операцій.

- 1) Завдання реалізувати в цілому за технологією ООП. Все оформити одним модулем (файлом).
- 2) 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>