NOM, Prénom :

Passage à la caisse

NSI2

On peut créer une classe `Caisse` qui va fonctionner un peu comme la classe `Ball` déjà rencontrée, avec

- Une variable **de classe** `nb_caisses` valant 0 au départ ;
- Une variable **de classe** `caisses` de type `list`, valant `[]` au départ, pour stocker les différentes caisses ;
- Une variable **de classe** `nb_clients_servis` valant 0 ;
- Une méthode `__init__` qui crée des instances de classes avec (au minimum) les attributs suivants :
  - `self.file`, la file d'attente, passée en paramètre dans les constructeur (ainsi dans les prochaines partie, chaque caisse pourra avoir sa propre file) ;
  - `self.temps_attente`, un `int` mesurant le nombre de tours avant que la caisse soit libre ;

  Quand le constructeur est appelé, `Caisse.nb_caisses` augmente de 1 et l'instance (`self`) est ajoutée à la liste `Caisse.caisses`.
- Une méthode `sert_client` qui
  - commence par enlever un client de sa file : alors on récupère son heure d'arrivée (notons la `heure`) dans la file et `tour-heure` nous donne son temps d'attente, qu'on peut ajouter à la variable `Caisse.temps_attente_total` ;
  - comme on sert un nouveau client, on peut incrémenter `Caisse.nb_clients_servis`
  - le temps passé à s'occuper du client est `randint(1,NB_CAISSES)` et devient la nouvelle valeur de l'attribut `temps_attente` de la caisse.
- Une méthode `actualise` qui sera plus tard appelée une fois par tour et :
  - enlèvera 1 au temps d'attente de la caisse ;
  - si elle est libre, servira un client (s'il y en a dans la file) ;

Boucle principale

On pourra créer `NB_CAISSES` caisses, créer une constante `NB_TOURS` et boucler sur la variable `tour` : tant qu'on a pas atteint `NB_TOURS`, on

- fait arriver un client dans la file ;
- parcourt la liste `Caisse.caisses` ;
- actualise chaque caisse ;
- termine en ajoutant 1 à `tour`

Fin du programme

C'est à vous de jouer, vous avez tout pour calculer le temps moyen d'attente par client servi.

Plusieurs files, au hasard

Adapter le programme précédent avec une file par caisse, avec 1 client par file au départ, et les suivants arrivent en choisissant une file au hasard sans en changer.

Plusieurs files, au hasard

Adapter le programme précédent avec une file par caisse, avec 1 client par file au départ, et les suivants arrivent en choisissant une caisse libre ou une avec le moins de monde.

Bilan

Dresser le bilan des 3 méthodes.