

# Logiciel de gestion de files d'attente

TP – Nathan BOSY – TG4

## Description :

Le logiciel est un gestionnaire de files d'attente qui a plusieurs fonctionnalités. Le nombre de clients AINSI que le nombre de services sont infinis. Chaque client choisi le service qu'il demande. Chaque service à sa propre liste d'attente. Le guichetier n'a plus qu'à entrer le service qui demande un nouveau client et une interface Tkinter permet d'afficher le nom du client demandé ainsi que le service appelé.

## Structure du code :

Le code est divisé en deux parties : Premièrement, nous avons une classe qui tout d'abord initialise le dictionnaire. Elle est composée de trois méthodes :

- La première ajoute un client dans une liste d'attente du service demandé
- La seconde permet au guichetier de chercher le prochain client d'un service
- La troisième permet, s'il reste des clients dans la file d'attente demandée, d'afficher à l'aide de Tkinter le nom du client ainsi que le service associé.

Secondement, nous avons une fonction permettant de choisir une méthode de cette classe. Il y a trois possibilités possibles :

- n : Permet d'ajouter un client avec f.gererClient()
- s : Demande le service afin de lancer l'interface appelant le prochain client
- q : Quitter le programme à l'aide d'assert.

Si aucune des trois solutions est entrée, le logiciel relance cette fonction en rappelant les différentes actions.

### Objectifs non atteints :

La prochaine étape de mon application serait de pouvoir l'utiliser entièrement à l'aide de Tkinter avec deux interfaces graphiques.

### Code avec commentaires :

```
'''
LOGICIEL DE GESTION DE FILES D'ATTENTE
    Créé par Nathan BOSY
'''

from tkinter import *

'''
Classe qui permet d'initialier le dictionnaire, avec toirs méthodes :
    1 - Ajoute un client dans une liste d'attente
    2 - Permet au guichetier de chercher le prochain client
    3 - Affiche avec Tkinter le client qui est demandé au service en question
'''
class Guichet:

    def __init__(self):

        print('    LOGICIEL DE GESTION DE FILES D\'ATTENTE\n',
              '(n = nouveau / s = suivant / q = quitter)\n')

        self.attente = {} # initialisation du dictionnaire

    def gererClient(self):

        print          ('\n\nÀ remplir côté client')
        nom =          input("Entrez votre nom : ").upper() # upper pour tout
mettre en majuscules
        service = input("Entrez le service souhaité : ").upper()

        if service in self.attente : # rajoute la valeur si le service existe
déjà
            self.attente[service].append(nom)

        else :
```

```

        self.attente[service] = [nom] # créé la clé du service s'il
n'existe pas

    def gereGuichetier(self):

        print          ('\n\nÀ utiliser côté Guichet')
        service = input("Entrez le service appelant un client : ").upper()

        if service in self.attente : # s'il reste du monde dans le service
demandé

            client = self.attente[service][0]
            self.attente[service].pop(0) # supprime le client du tableau

            if len(self.attente[service]) == 0 :
                self.attente.pop(service) # supprime la clé du service s'il ne
reste personne

            return client, service

        else : print('\n\nIl n\'y a personne en attente dans le service
demandé')

    def afficheur(self, texte):

        root = Tk()
        frame = Frame(root)

        nom = Label(frame, text=texte[0], font=("Arial", 40)) # nom de la
personne
        service = Label(frame, text=f'est attendu(e) au service
{texte[1]}', font=("Arial", 20)) # service demandé

        quitter = Button(root, text="Quitter ", command=root.destroy) #
bouton pour quitter l'interface
        quitter.pack(side=BOTTOM)

        root.geometry("800x300")
        frame.place(relx=.5, rely=.5, anchor=CENTER)

        nom.pack()
        service.pack()
        root.mainloop()

f = Guichet()

...
Fonction qui permet d'utiliser la classe en fonction de la méthode demandée

```

```

        n - Permet d'ajouter un client avec f.gererClient()
        s - Demande le service afin de lancer l'interface appelant le prochain
client
        q - quitter le programme
    ...
def programme() :
    entree = input("\n\nVeuillez choisir une action : ").lower()

    temp = 0

    if entree == "n" :
        temp = 1
        f.gererClient()

    elif entree == "s" :
        temp = 1
        b = f.gereGuichetier()

        if b != None : # seulement si f.gereGuichetier() nous renvoie un
client
            f.afficheur(b)

    assert entree != 'q', ('\n\nMerci d\'avoir utilisé nos services !') #
arrête le code si on entre "q"

    if temp != 1 :
        print('\n\nVeuillez entrer une action valide.',
            '\n(n = nouveau / s = suivant / q = quitter)\n')

while True :
    programme()

```