

Structure de données abstraites (SDA)

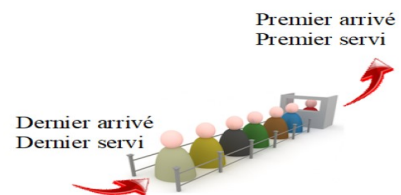
Structures séquentielles

Les Files

Définition : En informatique, une **file** (**queue** en anglais) est une structure de données linéaire basée sur le principe « Premier entré, Premier sorti », en anglais **FIFO (First In, First Out)**, ce qui veut dire que les premiers éléments ajoutés à la file seront les premiers à être récupérés.

Il faut se représenter une file comme... une file d'attente !

On ne peut entrer dans la file qu'en dernière position et on ne peut la quitter que si on est le premier. L'ajout d'un élément dans une file ne peut se faire qu'à la fin (en dernière position) et le retrait d'un élément ne peut se faire qu'au début (en première position)



Application :

En général, on utilise des files pour mémoriser temporairement des transactions qui doivent attendre pour être traitées :

- Les serveurs d'impression, qui doivent traiter les requêtes dans l'ordre dans lequel elles arrivent, et les insèrent dans une file d'attente (ou une queue en anglais).
- Certains moteurs multitâches, dans un système d'exploitation, qui doivent accorder du temps-machine à chaque tâche, sans en privilégier aucune.

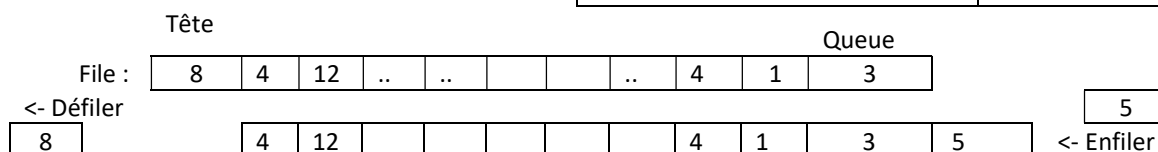
Représentation d'une file :

Une file contenant les éléments 'a','b','c' ('a' étant le premier et 'c' le dernier) sera représentée :

< 'a','b','c' <

Opérations sur les files

CREER_FILE () -> File() :	crée une file vide
ENFILER (f, element) -> None	ajoute l'objet element à la queue de la file f
DEFILER(f:File) -> elem	retire l'élément elem à la tête de f et retourne elem
EST_VIDE(f:File) -> Bool	renvoie True si la file f est vide sinon False



Exercice 1 : soit la suite d'instructions :

Représenter F à la fin ?

```
F=CREER_FILE()
ENFILER(F,21)
ENFILER(F,22)
ENFILER(F,23)
N=DEFILER(F)
ENFILER(F,24)
ENFILER(F,25)
N=DEFILER(F)
```

Exercice 2 :

```
f1 = File()
f2 = File()
f1.enfiler("a")
f1.enfiler(3)
x = f1.defiler()
f2.enfiler(x)
f1.enfiler("b")
f1.enfiler(6)
f2.enfiler(f1.defiler())
```

Représenter les files f1 et f2 à la fin de l'exécution :

Exercice 3 :

```
f3 = File()
f4 = File()
f5 = File()
for i in range(10):
    f3.enfiler(i)
for j in range(5):
    f4.enfiler(f3.defiler())
    f5.enfiler(f3.defiler())
```

Que contiennent f3,f4 et f5 à la fin ?

Implémentation d'une file (télécharger et réaliser le notebook : implémentation_files.ipynb)

Compléter le notebook avec les exercices suivants :

Exercice 4 : On dispose d'une file, écrire une fonction qui renvoie la file inversée (l'élément de la tête sera situé à la queue et ainsi de suite) On utilisera seulement les méthodes associées aux piles et aux files

Exercice 5 : Même question mais pour une pile

Exercice 5 : On dispose d'une file contenant des entiers, écrire une fonction qui renvoie une file où on aura séparé les nombres pairs des impairs

Exercice 6 : On dispose d'une pile contenant des entiers, écrire une fonction qui renvoie une pile où l'on aura séparé les nombres pairs des impairs