# UE ALGO5 — TD2 — Séance 8 : Files et FAPs

On souhaite programmer une structure de données destinée à gérer des requêtes d'impression envoyées à une imprimante. Pour des raisons matérielles, on impose que cette structure de données soit implémentée sans faire appel à un mécanisme d'allocation dynamique de la mémoire, c'est-à-dire uniquement à l'aide d'un (ou plusieurs) tableau(x) déclaré(s) statiquement au début du programme.

Les requêtes d'impression sont *déposées* par les utilisateurs dans une structure de donnée globale, de nom FileRequête, dans laquelle l'imprimante viendra *extraire* une tâche à effectuer. On souhaite dans un premier temps que les requêtes soient traitées selon une politique « premier arrivé, premier servi ».

Les primitives nécessaires à la gestion de `FileRequête` sont les suivantes :

Requête : un type
File : le type {*... à compléter ...*}
FileRequête : une File

Initialiser
{ *paramètres : aucun*
*valeur de retour : aucune*
*description : initialise FileRequête avec un contenu vide*
*effets de bord : FileRequête est modifiée* }

PlusDeRequête
{ *paramètres : aucun*
*valeur de retour : un booléen*
*description : vaut vrai ssi FileRequête est vide*
*effets de bord : aucun* }

Déposer
{ *paramètres : une requête x (donnée)*
*valeur de retour : aucune*
*description : ajoute x dans FileRequête*
*effets de bord : FileRequête est modifiée* }

Extraire
{ *paramètres : une requête x (résultat)*
*valeur de retour : aucune*
*description : FileRequête est non vide, supprime la requête la plus ancienne et la mémorise dans x*
*effets de bord : FileRequête est modifiée* }

Consulter
{ *paramètres : aucun*
*valeur de retour : une requête*
*description : FileRequête est non vide, renvoie la requête la plus ancienne*
*effets de bord : aucun* }

## Exercice 1.

- Discutez des différentes implémentations possibles du type File. Dans chaque cas, donnez une estimation du coût en temps d'exécution pour chacune des fonctions.
- Choisissez une des solutions possibles et implémentez chacune des fonctions.

**Exercice 2.** On souhaite maintenant associer une *priorité* à chaque requête : lors de l'exécution de la primitive Extraire c'est une requête (quelconque) de priorité maximale (parmi les requêtes présentes) qui doit être extraite. Le nombre de priorité sera supposé fini.

Modifiez la spécification des primitives, et reprendre les questions de l'exercice 1 dans ce nouveau contexte.