

UE ALGO5 — TD2 — Séance 6 : Files et files à priorité

Objectifs

À la fin de cette séance, vous devriez être capable de :

- manipuler et concevoir des files ou des FAP comme des structures abstraites ;
- réfléchir aux propriétés de files ou FAP ;
- proposer des implémentations de files et de FAP cohérentes avec les spécifications choisies.

Exercice 1.

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 de type *File*, 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 manipulation de valeurs du type *File* sont les suivantes :

<p>1 Requête : un type File : type { ... à compléter ... }</p> <p>3 Initialiser(F) 5 { Donnée-résultat : une File F Post-condition : F est initialisé avec un contenu vide 7 Effet de bord : F est modifiée }</p> <p>9 EstVide(F) 11 { Donnée : une File F Résultat : un booléen vrai ssi F est vide }</p> <p>13 Déposer(F,x) 15 { Donnée-résultat : une File F Donnée : une Requête x 17 Post-condition : x est ajoutée dans F Effets de bord : F est modifiée }</p>	<p>19 Extraire(F,x) 21 { Donnée-résultat : une File F Résultat : une Requête x 23 Pré-condition : F non vide Post-condition : supprime la requête la plus ancienne 25 de F et la mémorise dans x Effets de bord : F est modifiée }</p> <p>27 Consulter(F) 29 { Donnée-résultat : une File F Résultat : la Requête x la plus ancienne de F 31 Pré-condition : F non vide Effets de bord : aucun }</p> <p>33 35</p>
--	--

Q 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.

Q 2. 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.

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