

## TP4 : File d'attente avec priorité

L'objectif de ce TP est de réaliser la classe `SalleDAttenteAvecPrio` étudiée en TD 4, question 3, en lui affectant comme éléments des occurrences de la classe `Article` du TD 3. Concrètement, il s'agit de gérer une file d'attente d'articles en déclarant une variable de type `SalleDAttenteAvecPrio`. Vous aurez à rédiger un compte-rendu d'implémentation qui comprendra au moins le code source de la classe `SalleDAttenteAvecPrio`.

Les classes et interfaces suivantes sont fournies dans `/share/m2cci/PRG2/TP4` :

- Une classe d'application : `TestPrio`.
- Les interfaces `SalleDAttente` et `AvecPrio` et la classe `SalleDAttentePAPS` vues en TD 4.
- Les classes `Article` et `Livre` sont fournies. Il faudra modifier ces deux classes de manière à calculer la priorité de leurs instances suivant ces règles :
  - La priorité d'un article est la différence entre le `STOCK.MINIMAL` (à déclarer comme constante de classe) et la quantité de l'article ; on peut considérer cette valeur comme étant le degré d'urgence de réapprovisionnement. Si la quantité est supérieure ou égale à ce stock minimal, la priorité est égale à 0.
  - La priorité d'un livre se calcule de la même façon, tout en considérant un stock minimal augmenté de 50% par rapport à celui appliqué aux articles.

Seule la classe `SalleDAttenteAvecPrio` n'est pas fournie.

**Question 1 :** Finalisez la classe `SalleDAttenteAvecPrio` écrite en TD 4.

**Question 2 :** Ajoutez la méthode `reorganiser()`, appelée par la classe d'application lorsque l'utilisateur choisit cette quatrième option (il faudra alors décommenter l'appel de la méthode). La réorganisation a pour but de reconstruire les files d'attente par niveau de priorité à l'intérieur de l'objet `SalleDAttenteAvecPrio`, suite aux éventuels changements de quantité en stock (et donc de priorité) concernant des articles déjà enregistrés.

Le but de la méthode est donc de remettre à leur place les articles en fonction de leur priorité actuelle. La règle est la suivante : tout article ayant une priorité courante différente de celle de sa file d'attente courante, est enlevée de celle-ci, puis rajoutée en queue de la bonne file, c'est-à-dire celle correspondant à sa vraie priorité.