

PILE ET FILE BORNÉES

OBJECTIF : L'objectif de ce tp est de rendre l'élève capable :

- de créer une structure de pile bornée
- de créer une structure de file bornée à l'aide d'une liste **Python**
- de créer une structure de file bornée à l'aide d'un tableau

1 De la pile non bornée à la pile bornée

On souhaite établir une classe `Pile` dont la structure est celle d'une pile bornée. On se base alors sur la classe `Pile` utilisée dans le Tp précédent.

Q - 1 : Créer un fichier `pile_et_file_bornees.py` et l'importer dans un autre fichier avec l'alias `pfb`.

Pour l'aspect borné, il est possible d'introduire un attribut `_taille` à une instance de la classe.

Les seuls actions possibles avec la classe `Pile` sont :

- créer une pile `pile = Pile(taille)`
- tester si la pile est vide `pile.est_vide()`
- tester si la pile est pleine `pile.est_pleine()`
- empiler `pile.empiler(x)` pour ajouter `x` en haut de la pile. Ajouter un élément à une pile pleine conduit à un message d'erreur `overflow ou stackoverflow`
- dépiler `x = pile.depiler()` pour enlever le dernier élément de la pile et affecter sa valeur à `x`
- obtenir la valeur du sommet `x = pile.sommet()`

Q - 2 : Modifier la classe `Pile` du Tp précédent pour rendre la pile bornée.

2 Files bornées

2.1 De la pile bornée à la file bornée

Dans un premier temps, on élabore une structure de file bornée à l'aide d'une liste **Python** dont la tête sera toujours le premier élément et la queue, toujours le dernier.

Les seuls actions possibles avec la classe `File_L` sont :

- créer une file `file = File_L(taille)`
- tester si la file est vide `file.est_vide()`
- tester si la file est pleine `file.est_pleine()`
- enfiler `file.enfiler(x)` pour ajouter `x` en queue de la file. Ajouter un élément à une file pleine conduit à un message d'erreur.
- défiler `x = file.defiler()` pour enlever l'élément en tête de la file et affecter sa valeur à `x`.
- obtenir la valeur en tête `x = file.tete()` et la valeur de queue `x = file.queue()`

Q - 3 : Définir la classe *File_L*.

2.2 De la file bornée stockée dans une liste à la file bornée stockée dans un tableau

On élabore une structure de file bornée à l'aide d'un tableau. La tête de la liste est repérée par son indice `_i_tete` et la queue par l'indice de la première case libre dans le tableau `_i_queue`.

On associera donc 3 attributs à la file : `_taille`, `_i_tete` et `_i_queue` associés à la taille de la file, à l'indice de sa tête et à l'indice de sa queue.

Les seuls actions possibles avec la classe *File_T* sont :

- créer une file `file = File_T(taille)`
- tester si la file est vide `file.est_vide()`
- tester si la file est pleine `file.est_pleine()`
- enfiler `file.enfile(x)` pour ajouter `x` en queue de la file. Ajouter un élément à une file pleine conduit à un message d'erreur.
- défiler `x = file.defile()` pour enlever l'élément en tête de la file et affecter sa valeur à `x`.
- obtenir la valeur en tête `x = file.tete()` et la valeur de queue `x = file.queue()`
- obtenir l'indice de la tête `i = file.indice_tete()` et l'indice de la queue `i = file.indice_queue()`

Q - 4 : Définir la classe *File_T*.