



Sujet Stack TDD





Enoncé

Créer une class `MyStack` contenant 2 listes d'entier en attributs qui sont en private (`stack_a`; `stack_b`).

Elle aura les fonctions public :

- 📌 `front_a()` qui renverra le premier element de la liste a
- 📌 `back_a()` qui renverra le dernier element de la liste a
- 📌 `front_b()` qui renverra le premier element de la liste b
- 📌 `back_b()` qui renverra le dernier element de la liste b



Enoncé

 sa() swap les 2 premiers elements de la liste a

```
Stack stackA= new Stack([1, 2, 3, 4, 5]); // exemple n'est la que pour imager le fonctionnement  
stackA.sa();  
// stackA = [2, 1, 3, 4, 5]
```

 sb() swap les 2 premiers elements de la liste b



Enoncé

Il faudra intégrer les fonctions `pa()` et `pb()` qui prennent le premier élément de la liste et le met au dessus de la liste b ou a.

ps : pa permettra de prendre le premiere elements de la stack a pour le mettre au top de la stack b et inversement pour pb



Exemple de PA

```
stack_a = Stack([1, 2, 3])
stack_b = Stack([4, 5, 6])

stack_a.pa()
print(stack_a.elements) # affichera [2, 3]
print(stack_b.elements) # affichera [1, 4, 5, 6]
```



Fonction `ra()` `rb()`

Il faudra intégrer les fonctions `ra()` et `rb()` qui font une rotation de la liste.

Le premiere elements devients le dernier et le second devient le premier ect...

```
stack = Stack([1, 2, 3])
stack.ra()
print(stack.elements) # affichera [2, 3, 1]
```



Fonction rra() rrb()

Il faudra intégrer les fonctions `rra()` et `rrb()` qui font une rotation inverse de la liste.

Le dernier elements devient le premier et le second le second devient le troisieme...

```
stack = Stack([1, 2, 3])
stack.rra()
print(stack.elements)  # affichera [3, 1, 2]
```



Exercice

Vous commencerez par écrire les tests unitaires pour chaque fonction.

Les tests devront être le plus complet possible.

Vous devrez ensuite écrire le code pour que les tests passent.

Pour le bien de l'exercice, faites un constructeur qui permet d'initialiser vos deux listes.



Exemple de cas de test

- 📌 Cas 1 : test avec stack de 3 [2,1,3]
- 📌 cas 2 : test avec un seul element [2]
- 📌 cas 3 : 10 elements randoms
- 📌 cas Stack : Si stack vide§



Exercice 1.2

Ajouter une fonction `rr()` qui fait une rotation des deux listes ainsi que `rrr()` qui reverse rotate les deux listes.



Exercice 2

Vous devrez ensuite changer le type de la list en `int []` pour la partie refactor.



Exercice Final

Faire un programme qui prend en paramètre une liste de nombre et qui les trie par ordre croissant en utilisant les fonctions de la class

`MyStack`.