

5.4

Implémentation des piles

NSI TERMINALE - JB DUTHOIT

5.4.1 Implémenter une pile avec un tableau

 **Exercice 5.119**

Une pile bornée est une pile dotée à sa création d'une capacité maximale. On se propose l'interface suivante :

```
creer_pile(c) #crée et renvoie une pile de capacité c
est_vide(pile) # Renvoie True si pile est vide
est_pleine(pile) #Renvoie True si pile est pleine
empiler(pile,elt) #empile elt
depiler() #dépile
```

Pour réaliser cette pile, on utilisera un tableau dont la taille est fixée à la création et correspond à sa capacité.

5.4.2 Implémenter avec un tableau dynamique

L'implémentation des piles en python se fait facilement à l'aide des méthodes `append()` et `pop()` du type `list` :

```
ma_pile.append(ma_valeur) # permet d'empiler une valeur
ma_pile.pop() # permet de dépiler une valeur
len(ma_pile) # renvoie la longueur de ma_pile
```

 **Exercice 5.120**

Implémenter une pile avec les tableaux dynamiques

 **Exercice 5.121**

En utilisant des tableaux dynamiques, construire l'objet `Pile`. Cette objet disposera d'une interface classique (création d'une pile vide, vérification si la pile est vide, empilage et dépilage)

5.4.3 Implémenter une pile avec des objets

 **Exercice 5.122**

On considère la classe `Cellule` suivante :

```
class Cellule:
    '''Une cellule d'une liste chaînée'''
    def __init__(self,v,s):
        self.valeur = v
        self.suivante = s
```

Construire la classe `Pile` qui permettra d'instancier une pile. (cf le type abstrait de la pile pour les

méthodes)

5.4.4 Utiliser une pile

Exercice 5.123

Revisiter la classe `Pile` en lui ajoutant un attribut `_taille` indiquant à tout moment la taille de la pile. Quelles méthodes doivent être modifiées. Proposer une méthode `get_taille` qui renvoie la taille de la pile.

Exercice 5.124

On considère une chaîne de caractères composée de parenthèses ouvrantes "(" et de parenthèses fermantes ")".

Une chaîne est bien parenthésée si chaque ouvrante est associée à une unique fermante, et réciproquement. Écrire une fonction `parenthese(chaine)` prenant en paramètre une chaîne de caractère composée de parenthèses ; La fonction renvoie `True` si la chaîne est bien parenthésée, `False` sinon.

Exercice 5.125

Cet exercice est un prolongement du précédent.

Écrire une fonction `parenthese(chaine)` prenant en paramètre une chaîne de caractères ; La fonction renvoie `True` si la chaîne est bien parenthésée, `False` sinon.

Par exemple, `parenthese('(1+2*3)*4')` va renvoyer `True`.

Exercice 5.126

Cet exercice est un prolongement du précédent.

Écrire une fonction `parenthese(chaine)` prenant en paramètre une chaîne de caractères (la chaîne de caractère contiendra ici, en autres des parenthèses "(", ")" et des crochets "[", "]") ; La fonction renvoie `True` si la chaîne est bien parenthésée, `False` sinon.

Par exemple, `parenthese('((1+2*3)*4)')` va renvoyer `True`.

Exercice type BAC 5.127

Session juin 2021 - Metropole -EXercice 2

➔ Cet exercice traite des notions de piles et de programmation orientée objet.