

5.4

Implémentation des piles

NSI TERMINALE - JB DUTHOIT

5.4.1 Implémenter avec un tableau dynamique python

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 5.118

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 une pile avec des classes

Exercice 5.119

Implémenter une pile avec une classe !

5.4.3 Implémenter une pile avec des listes chaînées

Exercice 5.120

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 5.121

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 5.122

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 5.123

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 5.124

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`.