

# Chapitre 14 – Les piles

---

Structures séquentielles.

Une pile est un cas particulier de liste : les insertions et les suppressions se font à une seule extrémité, appelé *sommet de la pile*. LIFO : Last In First Out.

# TAD Pile

---

Sorte : Pile

Utilise : Élément, Booléen

Opérations :

Créer-pile-vidé() → Pile {constructeur}

Empiler(Pile, Élément) → Pile {constructeur}

Dépiler(Pile) → Pile {constructeur}

Sommet(Pile) → Élément {observateur}

Pile-vidé(Pile) → Booléen {observateur}

# Représentation contiguë (par tableau)

---

- Plus de décalage par rapport aux listes quand on empile ou dépile. Ces opérations se font toujours sur le 1<sup>er</sup> élément. Sommet indique à tout instant la position du 1<sup>er</sup> élément de la pile.

```
Type pile = Enregistrement  
    t : tableau [1..n] d'élément  
    sommet : entier  
    FinEnregistrement  
Var p : pile
```

L'implémentation du type pile est triviale.

# Représentation chaînée 1/2

---

Les suppressions et les insertions se font toujours en tête de liste.

```
Type    pile= ^cellule
         cellule = Enregistrement
                   val : élément
                   succ : pile
         FinEnregistrement
```

```
Var p : pile
```

# Représentation chaînée 2/2

---

Procédure empiler (E e : élément, E/S p : pile)

Var q : pile

Début

q ← allouer(cellule)

q^.val ← e

q^.succ ← p

p ← q

Fin

Procédure dépiler (E/S p : pile)

Var q : pile

Début

q ← p

p ← p^.succ

recupérer(q)

Fin