

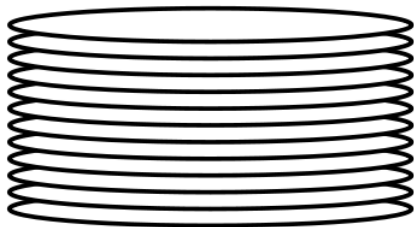
# Structures de données

Listes chaînées - Piles et Files - Arbres - Graphes

M. Tellene

# Structures de données - Piles

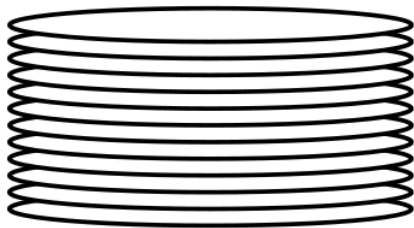
Le concept de pile est assez connu. Imaginez une pile d'assiette à nettoyer



Comment laver les assiettes une par une ?

# Structures de données - Piles

Le concept de pile est assez connu. Imaginez une pile d'assiette à nettoyer



Comment laver les assiettes une par une ?

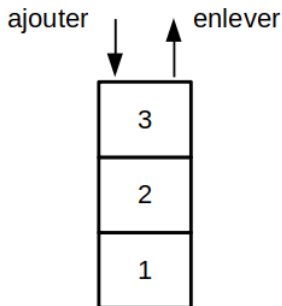
En prenant l'assiette au sommet, la lavant et recommencer tant qu'il reste des assiettes

# Structures de données - Piles

C'est le concept du « dernier entré, premier sorti » ou en anglais **LIFO** pour Last In First Out

En termes de structure de données, une pile est une séquence de  $x_0, x_1, \dots, x_{n-1}$  de valeurs où il est possible de retirer et d'ajouter un élément du **même côté de la séquence**

Ainsi dans une pile, seul le sommet est accessible



# Structures de données - Piles

## Méthodes de la classe Pile

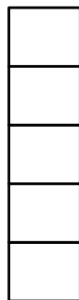
- constructeur `Pile()`
  - Postcondition : on crée un attribut contenu initialisé avec un tableau vide
- Méthode `empiler(x)`
  - Postcondition : l'élément `x` est ajouté en sommet de la pile
- Méthode `depiler()`  $\rightarrow$  tous types
  - Précondition : la pile n'est pas vide
  - Postcondition : le sommet de la pile est dépilé (enlevé de la pile) et retourné
  - Résultat : renvoie le dernier élément de la file
- Méthode `vider()`
  - Postcondition : la pile ne contient plus aucun élément
- Méthode `est_vide()`  $\rightarrow$  bool
  - Résultat : renvoie `True` si la pile est vide, `False` sinon

# Structures de données - Piles

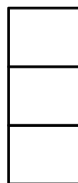
Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- Étape 3 : +
- Étape 4 : 1
- Étape 5 : )
- Étape 6 : \*
- Étape 7 : 5



P1



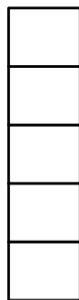
P2

# Structures de données - Piles

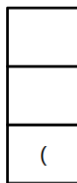
Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- **Étape 1 :** (
- Étape 2 : 2
- Étape 3 : +
- Étape 4 : 1
- Étape 5 : )
- Étape 6 : \*
- Étape 7 : 5



P1



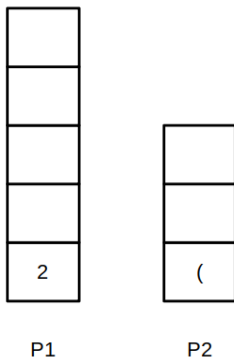
P2

# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- **Étape 2 : 2**
- Étape 3 : +
- Étape 4 : 1
- Étape 5 : )
- Étape 6 : \*
- Étape 7 : 5



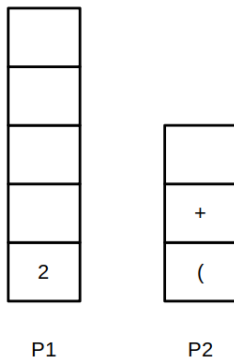


# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- **Étape 3 : +**
- Étape 4 : 1
- Étape 5 : )
- Étape 6 : \*
- Étape 7 : 5

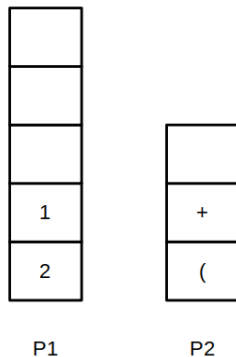


# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- Étape 3 : +
- **Étape 4 : 1**
- Étape 5 : )
- Étape 6 : \*
- Étape 7 : 5

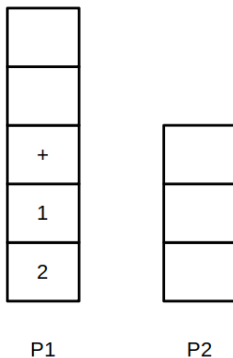


# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- Étape 3 : +
- Étape 4 : 1
- **Étape 5 : )**
- Étape 6 : \*
- Étape 7 : 5

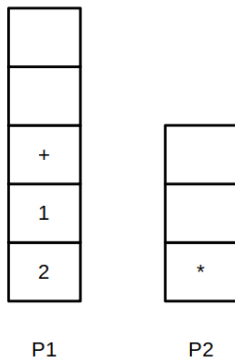


# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- Étape 3 : +
- Étape 4 : 1
- Étape 5 : )
- **Étape 6 : \***
- Étape 7 : 5

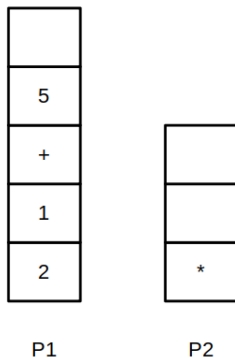


# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- Étape 3 : +
- Étape 4 : 1
- Étape 5 : )
- Étape 6 : \*
- **Étape 7 : 5**

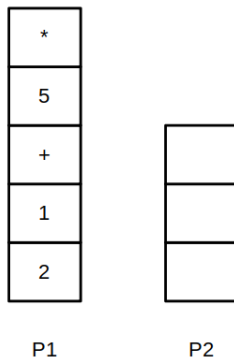


# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

- Étape 1 : (
- Étape 2 : 2
- Étape 3 : +
- Étape 4 : 1
- Étape 5 : )
- Étape 6 : \*
- Étape 7 : 5



# Structures de données - Piles

Exemple d'utilisation d'une pile : l'évaluation d'expression arithmétique

Soit l'expression suivante :  $(2 + 1) * 5$

Ainsi, les expressions  $(2 + 1) * 5$  et  $2 + 1 * 5$  sont équivalentes

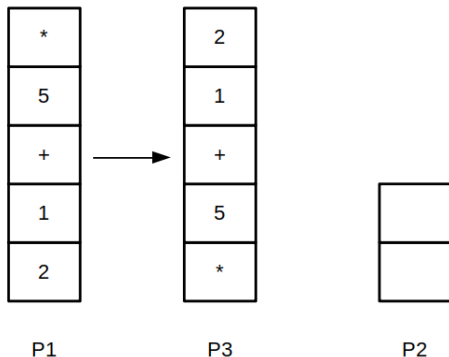
Plus précisément la notation  $(2 + 1) * 5$  est appelé notation infixée classique (l'opérateur est entre les paires d'opérandes)

La notation  $2 + 1 * 5$  est quant à elle appelé notation postfixée (les paires d'opérandes sont suivies par l'opérateur)

# Structures de données - Piles

Une fois que nous avons la pile d'évaluation, nous pouvons calculer l'expression

- Étape 1
- Étape 2
- Étape 3
- Étape 4
- Étape 5

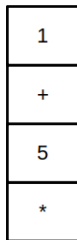




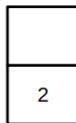
# Structures de données - Piles

Une fois que nous avons la pile d'évaluation, nous pouvons calculer l'expression

- **Étape 1**
- Étape 2
- Étape 3
- Étape 4
- Étape 5



P3

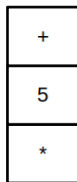


P2

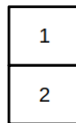
# Structures de données - Piles

Une fois que nous avons la pile d'évaluation, nous pouvons calculer l'expression

- Étape 1
- **Étape 2**
- Étape 3
- Étape 4
- Étape 5



P3



P2

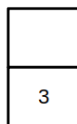
# Structures de données - Piles

Une fois que nous avons la pile d'évaluation, nous pouvons calculer l'expression

- Étape 1
- Étape 2
- **Étape 3**
- Étape 4
- Étape 5



P3



P2

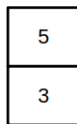
# Structures de données - Piles

Une fois que nous avons la pile d'évaluation, nous pouvons calculer l'expression

- Étape 1
- Étape 2
- Étape 3
- **Étape 4**
- Étape 5



P3



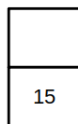
P2

# Structures de données - Piles

Une fois que nous avons la pile d'évaluation, nous pouvons calculer l'expression

- Étape 1
- Étape 2
- Étape 3
- Étape 4
- **Étape 5**

P3



P2

# Structures de données - Piles

Comparaison de complexité (dans le pire des cas) entre les tableaux, les listes chaînées et les piles

Structure de données	Accès	Recherche	Insertion	Suppression
Tableaux	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Listes chaînées	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Piles	$O(n)$	$O(n)$	$O(1)$	$O(1)$

$n$  représente le nombre d'éléments dans la structure de données

# Structures de données - Files

Tout comme les piles, le concept de file est assez connu.  
Imaginez une file d'attente à un guichet



Comment les personnes entrent et sortent de la file ?

# Structures de données - Files

Tout comme les piles, le concept de file est assez connu.  
Imaginez une file d'attente à un guichet



Comment les personnes entrent et sortent de la file ?

Ils arrivent par un côté et ressortent par l'autre



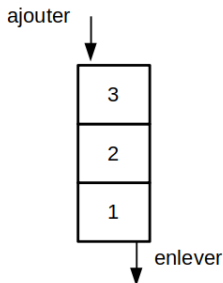
# Structures de données - Files

C'est le concept du « premier entré, premier sorti » ou en anglais **FIFO** pour First In First Out

Tout comme une pile, une file est une séquence de  $x_0, x_1, \dots, x_{n-1}$  de valeurs.

Au contraire de la pile, l'ajout d'éléments se fait par **un côté de la séquence** et le retrait se fait par **l'autre côté**

Ainsi dans une file, le sommet et la fin sont accessibles



# Structures de données - Files

## Méthodes de la classe File

- constructeur `File()`
  - Postcondition : on crée un attribut contenu initialisé avec un tableau vide
- Méthode `enfiler(x)`
  - Postcondition : l'élément `x` est ajouté en sommet de la file
- Méthode `defiler()` → tous types
  - Précondition : la file n'est pas vide
  - Postcondition : le dernier élément de la file (cf. le premier élément enfilé) est défilé (enlever de la file)
  - Résultat : renvoie le dernier élément de la file
- Méthode `vider()`
  - Postcondition : la file ne contient plus aucun élément
- Méthode `est_vide()` → bool
  - Résultat : renvoie `True` si la file est vide, `False` sinon

# Structures de données - Files

Exemple d'utilisation d'une file : le jeu de la bataille

La bataille est un jeu de cartes qui se joue habituellement à deux. Les deux joueurs se partagent l'ensemble d'un jeu de cartes (52 ou 32), qui n'en prennent pas connaissance. À chaque tour, chaque joueur retourne la carte du haut de sa main (ou son tas). Celui qui a la carte de la plus haute valeur fait la levée, qu'il place sous son tas.

En cas d'égalité de valeurs — cas appelé bataille — les joueurs commencent par placer une première carte face cachée puis une seconde carte face visible pour décider qui fera la levée. En cas de nouvelle égalité, la procédure est répétée. À la fin, le joueur gagnant remporte toutes les cartes, qu'il place sous son tas.

# Structures de données - Files

Exemple d'utilisation d'une file : le jeu de la bataille

→ Les paquets des joueurs sont symbolisés par des files

- les cartes **entrent** dans le tas dans un sens
- les cartes **sortent** du tas dans l'autre sens

# Structures de données - Files

Exemple d'utilisation d'une file : le jeu de la bataille

→ Les paquets des joueurs sont symbolisés par des files

- les cartes **entrent** dans le tas dans un sens
- les cartes **sortent** du tas dans l'autre sens

Si l'on veut voir plus loin, les cas de bataille peuvent être symbolisés par des piles

# Structures de données - Files

Exemple d'utilisation d'une file : le jeu de la bataille

# Structures de données - Files

Comparaison de complexité (dans le pire des cas) entre les tableaux, les listes chaînées et les piles

Structure de données	Accès	Recherche	Insertion	Suppression
Tableaux	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Listes chaînées	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Piles	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Files	$O(n)$	$O(n)$	$O(1)$	$O(1)$

$n$  représente le nombre d'éléments dans la structure de données