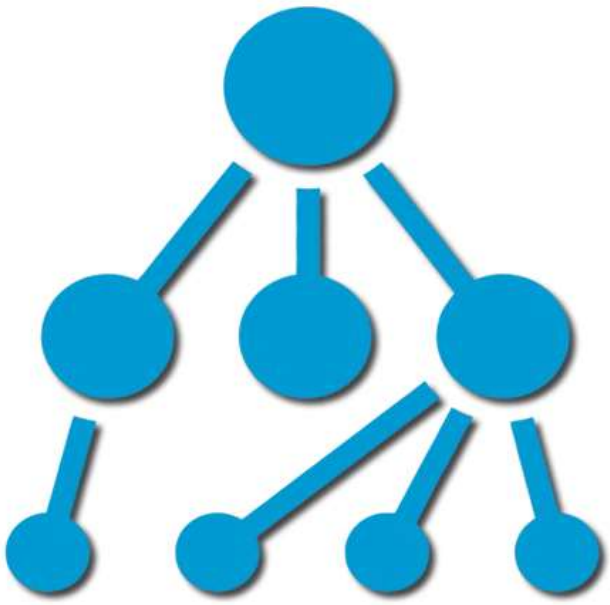
STRUCTURES DE DONNÉES

NSI 01



PILES, FILES

[Frédéric Peurière – Marion Szpieg]

1. Introduction sur les structures de données

Définition : un **type abstrait** ou une **structure de données abstraite** est une spécification mathématique d'un ensemble de données et de l'ensemble des opérateurs associés. On dit que ce type de données est abstrait car il correspond à un cahier des charges qu'une structure de données qu'il est ensuite possible d'implémenter.

On peut donc détailler une structure de données précise sans écrire aucune ligne de code : on appelle cela son **interface**. Une fois que l'interface est bien définie, on peut alors la programmer en Python (ou un autre langage de programmation) : on appelle cela **l'implémentation** de la structure de données.

Voici quelque avantages à la séparation entre l'interface et l'implémentation :

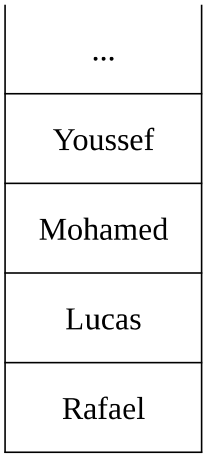
2. Les piles

2.1. Qu'est-ce que c'est ?

La structure de données linéaire «pile» est une suite ordonnées d'éléments qu'on construit avec le principe du « dernier arrivé, premier servi ». En informatique, ce principe de construction est communément appelé LIFO (car « Last In, First Out »). Une pile peut-être vide.

« Dans la vraie vie », on retrouve cette structure

Exemples :



En informatique, ce type de structure de données est souvent utilisée. Par exemple :

2.2. Interface d'une pile

Voici l'interface minimale d'une pile (à connaître !!) :

- `creer_p()` :
- `estVide_p(P)` :
- `empiler(P,e)` :
- `depiler(P)` :
- `sommet(P)` :
- `base(P)` :

Exemple : voici une suite de commandes :

```
1  p=créer_p()
2  p=empiler(p,'Francisco')
3  p=empiler(p,'Lucas')
4  p=empiler(p,'Youssef')
5  p=depiler(p)
6  p=empiler(p,'Mohamed')
7  p=empiler(p,Rafael')
8  p=depiler(p)
9  p=empiler(p,'Frédéric')
10 p=depiler(p)
```

Faire une représentation de la pile après la 5e commande, puis après la 10e commande :

À partir des fonctions primitives, l'interface d'une pile est améliorable par d'autres fonctions, par exemple :

- `longueur_p(P)` :
- `dernier_p(P)` :
- `inserer_p(P,e,i)` :
- `ieme_p(P,i)` :
- `remplacer_p(P,e,i)` :
- `supprimer_p(P,i)` :

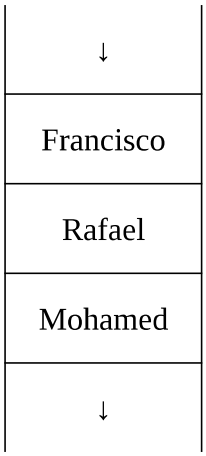
3. Les files

3.1. Qu'est-ce que c'est ?

La structure de données linéaire «file» est une suite ordonnées d'éléments qu'on construit avec le principe du «premier arrivé, premier servi ». En informatique, ce principe de construction est communément appelé FIFO (car «First In, First Out »). Une file peut-être vide.

Cette structure de données illustre le principe de la file d'attente à un guichet.

Exemple :



En informatique, ce type de structure de données est souvent utilisée. Par exemple :

3.2. Interface d'une file

Voici l'interface minimale d'une file (à connaître !!) :

- `creer_f()` :
- `estVide_f(F)` :
- `enfiler(F,e)` :
- `defiler(F)` :
- `début(F)` :
- `suite(F)` :

Exemple : voici une suite de commandes :

```
1  f=créer_file()
2  f=enfiler(f,'Frédéric')
3  f=enfiler(f,'Mohamed')
4  f=enfiler(f,'Rafael')
5  f=defiler(f)
6  f=enfiler(f,‘Francisco’)
7  f=enfiler(f,Lucas')
8  f=defiler(f)
9  f=enfiler(f,'Youssef')
10 f=defiler(f)
```

Faire une représentation de la file après la 5e commande, puis après la 10e commande :

Comme avec les piles, à partir des fonctions primitives, l'interface est améliorable par d'autres fonctions.