

exple d'usage des files : buffer
la pile est bcp utilisée en théorie des langages
parenthésage des expressions

pile : cf magic the gathering (jeu)

parenthésage : on m'a q ([ds la pile

la pile est la bonne structure de données pour résoudre
le pb donné (m's' il y a d'8 solutions)

pr 1 pb algo ? il faut se poser la question de la structure
de données à utiliser

ds l'endroit qq de la chaîne de caractères, on a

uniq^t besoin d'un ([ouvrant précédent

avec les élèves, il faudrait identifier ce dont on a
besoin : ici la dernière chose ouverte

ce qui induit l'utilisa^t de la structure de pile
en utilisant une liste on a accès à l'avantage
d'info et ça peut créer des erreurs

exple de pile : CTRL-Z les commandes ont été empilées
on peut les dépiler

le len() m'a pas été mis par défaut ds les structures
piles ou files car on m'a besoin q de savoir
si vide ou non, on traite ttq mon vide
on m'a accès qu'au 1^{er} el^{mt}

en pratique la méthode len peut être utilisée / surchargée
si on en a besoin pr résoudre 1 pb donné

le paradigme objet

/ struct 2

destructeur | terme général pr détruire un objet
et ses sous-objets
libère

ici il doit détruire les Node
ms en Python on ne manipule pas la mémoire explicitement
à le C ou le C++

en python le moteur gère la mémoire
on met tout ce à quoi réfère l'objet à None
et le garbage collector / ramasse miette fait le nettoyage
les 3 attributs de la liste sont à garbage collector

pr détruire : les cellules là

a = Obj()

b = Obj()

L = [a, b]

L = []

a = 1 & l'objet sur lequel pointait a passe à la poubelle

ici l'idée est de gérer nous m le fait de savoir
gérer qd on n'a plus besoin de qq chose

il faut mettre à jour ttes les ref de ts les élt avant de
faire le del ↳ vers l'elt à
supprimer

il faut d'abord q plus personne ne pointe vers l'elt
à supprimer

indice : pr inverser une file, on n'a pas besoin de
savoir c' elle a été implémentée de plutôt extérieur à la
classe

éviter les for elt in liste et la discussion sur les
itérateurs avec les élèves

progression term

commencer par la POO avec des cas ts simples
 classe date couleur point 2D / vecteur 2D/3D
 - jour - rouge - abs
 - mois - vert - ord
 - année - bleu

liste
 pile
 file
 arbre
 graphe

dans cet ordre

pile et file implémentées par liste
 plus utilisées dans parcours

graphe = arbre avec des cycles

pour les piles et files on utilise la classe liste qu'on a déjà
 et qui se comporte presq à on veut
 liste python ou liste dbt chaîne codée auparavant

q pile et file héritent de liste n'est pas une bonne idée
 pile et file seraient des listes particulières
 toutes les fonctionnalités de liste seraient disponibles pile/file
 à accéder au dernier élément ce qu'on ne veut pas pour une pile

on veut restreindre
 on prend un attribut de la classe liste
 et on autorise les opérations attendues
 on restreint l'utilisation de l'attribut de la classe liste
 on s'interdit d'y accéder directement
 on ne doit s'autoriser q de passer par les méthodes
 membres de la pile
 l'attribut de vrait s'appeler -- liste pour signifier
 qu'il est privé à la classe et ne doit pas être
 utilisé de l'ext. c'est une convention car
 en python tt est public

ds pile / file on n'ira pas chercher / appeler

/ struct 4

les attributs premier ou autre de la classe liste
dernier

on n'utilise q les méthodes de la classe liste

la plupart s'écrivent en 1 seule instruction qui
consiste à appeler la bonne instruction avec le bon paramètre

une pile est une encapsulation d'1 liste (avec des
contraintes d'utilisation)

destructeur : on surcharge avec le def

le garbage collector appelle le destructeur
celui qu'on a surchargé
ou le destructeur par défaut

on peut créer des attributs ailleurs q ds le
constructeur, mais il n'existeront q pr les obj
sur lesquels la fc^{te} contenant la construction
de cet objet est appelée

mettre un print ds le def pr voir à quel moment
les choses sont supprimées

ce n'est pas le code du destructeur qui
détruit l'obj. l'obj est détruit lorsqu'on sort
du destructeur (le def -- def --)

ds vider pas un def self

le but n'est pas de supprimer la liste de la mémoire
on pourrait ajouter un elt après vider
vider est seulement revenir à un objet ds l'état
qu'il avait à l'appel de init

puis le destructeur peut appeler vider

pas de del self à la fin du del
car c'est fait par défaut (rien ne surcharge pas)

self fait référence à | l'obj
| l'instance de la classe |

sur lequel on appelle la méthode

self l'objet

self. l'appel à un attribut ou une méthode sur cet obj

1 liste et 2 ref vers premier et dernier
des ref vers 2 cellules

1 cellule est 1 donnée info

2 refs précédent et suivant vers d'autres cellules

c le fait q les cellules soient chaînées qui permet
d'accéder à ttes

ms l'obj liste ne réfère q la 1^{ère} liste (simple chaînée)
2 attributs (en 1^{er} chaîne)

traiter retourne un obj et le supprime

fait plus choses, ça prend plus d'instruc^o à écrire

→ récupérer l'obj avant de l'enlever de la structure

nécessite 3 inst - récupère le sommet de pile/fila
- le supprime
- le retourne