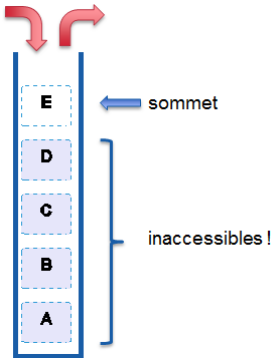
Algorithmique
Les piles

1 Qu'est-ce que c'est ?

Les piles sont des structures séquentielles appelées **LIFO** (**L**ast **I**n **F**irst **O**ut). C'est à dire que les entrées et les sorties s'effectuent du même coté.



— Le seul élément accessible est celui du dessus : le *sommet*, qui n'existe pas si la pile est vide.

— On ne peut que supprimer le sommet : l'opération *dépiler*. Il est impossible de dépiler une pile vide.

— Lorsqu'un élément est ajouté, il est placé "au-dessus" et devient donc le nouveau sommet : l'opération *empiler*.

2 Le type abstrait

Les opérations *sommet* et *dépiler* sont des opérations partielles, puisqu'elles ne peuvent pas être appliquées à une pile vide.

Comme toujours, il est nécessaire d'avoir une opération d'initialisation : *pilevide* et une pour vérifier si la structure contient des éléments : *estvide*.

Le type abstrait définissant les piles est donc le suivant :

SORTE
Pile

UTILISE
Booléen, Élément

OPÉRATIONS

pilevide	:	$\rightarrow$ Pile
estvide	:	Pile $\rightarrow$ Booléen
empiler	:	Pile $\times$ Élément $\rightarrow$ Pile
dépiler	:	Pile $\rightarrow$ Pile
sommet	:	Pile $\rightarrow$ Élément

PRÉCONDITIONS
dépiler(p) **est-défini-ssi** *estvide*(p) = faux
sommet(p) **est-défini-ssi** *estvide*(p) = faux

AXIOMES
dépiler(*empiler* (p, e)) = p
sommet(*empiler* (p, e)) = e
estvide(*pilevide*) = vrai
estvide(*empiler*(p, e)) = faux

AVEC
p : Pile
e : Élément

3 À quoi ça sert ?

— Inverser l'ordre des éléments ;
— évaluer des expressions ;
— simuler la pile des appels (pour rendre itératif des algorithmes récursifs)...