V.2 La Pile

Une *Pile* est un type permettant de représenter un ensemble de données, du même `type T`, à accès séquentiel et écriture non destructive.

L'accès aux données se fait dans l'ordre inverse duquel elles ont été insérées.

Définition : *Pile*

On appelle Pile, un ensemble formé d'un nombre variable de données sur lequel on effectue les opérations suivantes :

- ajouter une nouvelle donnée ;
- consulter la dernière donnée ajoutée et non supprimée depuis ;
- supprimer la dernière donnée ajoutée et non supprimée depuis ;
- savoir si l'ensemble est vide ou non ;

Fonctionnalités

```
init: Int → StackT // crée une pile (vide)
top: StackT → T | Vide // retourne le sommet de la pile
pop: StackT -> StackT x T // retire le sommet de la pile
push: StackT x T -> StackT // ajoute un nouvel élément
isEmpty: StackT -> Bool // vérifie si la pile est vide
isFull: StackT -> Bool // vérifie si la pile est pleine
capacity: StackT -> Int // nombre maximum d'éléments
count: StackT -> Int // nombre d'éléments dans la pile
```

Caractéristiques (features)

S1: isEmpty(init(n)) == true

S2: count(init(n)) == 0

S3: top(push(p,t))==t

S4: top(p)==Vide $\iff$ isEmpty(p)

S5: (p_2,t_2)=pop(push(p_1,t_1)) $\iff$ (p_2==p_1) && (t_2==t_1)

S6: (p_2,t)=pop(p_1) $\Rightarrow$ push(p_2,t) == p_1

S7: count(push(p,t)) == count(p)+1

S8: (p_2,t) = pop(t_1) $\Rightarrow$ count(p_2) == count(p_1) - 1

S9: pop(p) => !isEmpty(p)

S10: isFull(p) $\iff$ count(p) == capacity(p)

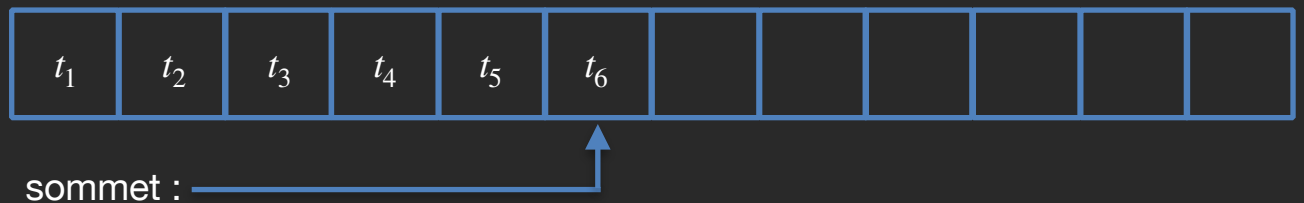
S11: isEmpty(p) $\iff$ count(p) == 0

S12: capacity(init(n)) == n

S13: push(p,t) $\Rightarrow$!isFull(p)

Description logique

Puisqu'il faut stocker des éléments, on peut utiliser un tableau. En revanche on doit interdire l'accès direct aux éléments, cet accès ne peut se faire que par les fonctions pop et push. Il faut donc être capable de connaître l'index du sommet de la pile.



s7 et s8 montrent que `count` et `top` sont fortement liés, il n'est donc pas nécessaires de les stocker tous les deux.

StackT : (storage: [T], count: Int)

Exercice :

Écrivez le protocol swift décrivant le type abstrait StackInt

Exercice :

Proposez une implémentation concrète du type abstrait StackInt.