

## Exercici pila amb closures:

Construir un *creador de Piles* amb *closures* (no una classe Pila, que seria el més raonable), igual que hem fet amb l'exemple del comptador. És a dir, feu una classe **Pila** amb un mètode de classe (boto **Class** del *Browser* premut) **creaPila** que retorni **sis blocs** en un **Array**, un per a cada operació típica de les piles: **reset pila**, **pop**, **top**, **push**, **mida**, i **buida?**.

Ho farem servir així (al *Playground*):

Primer definim

```
| p rst pop top push size empty |
```

```
p := Pila creaPila.
```

```
rst    := p at: 1.  
pop    := p at: 2.  
top    := p at: 3.  
push   := p at: 4.  
size   := p at: 5.  
empty  := p at: 6.
```

Després podem executar (encara al *Playground*), per exemple:

```
rst value.  
push value: 1.  
push value: $a.  
push value: 3.  
push value: 'a'.  
push value: 5.
```

```
top value traceCr.  
pop value.  
size value traceCr.  
empty value traceCr.
```

```
top value traceCr.  
pop value.  
size value traceCr.  
empty value traceCr.
```

```
top value traceCr.  
pop value.  
size value traceCr.  
empty value traceCr.
```

```
pop value.  
pop value.  
empty value traceCr.
```

El resultat al **Transcript** hauria de ser:

```
5
4
false
'a'
3
false
3
2
false
true
```

Obviament **així NO és com es fa una pila en Smalltalk!!** Però és un exercici amb *closures* que val la pena pensar.

Pistes: Podeu utilitzar per a la pila una **OrderedCollection** i afegir i treure elements per un sol costat, p.ex. pel final. Així, podeu fer servir **#addLast:**, **#last**, **#removeLast**, etc.