```
Object subclass: #Stack
    instanceVariableNames: 'count elements'
    classVariableNames: 'MaxDepth'
    poolDictionaries: ''
    category: 'Example'
!

!Stack class methodsFor: 'creation'!
initialize
"sets default depth"
    MaxDepth <- 100

new
"builds a new stack of default depth"
    ^ super new init: MaxDepth
!
new: desiredDepth
"builds a new stack of given depth"
    ^ super new init: desiredDepth
!!
```

```
!Stack methodsFor: 'initialization'!
init: depth
    count <- 0.
    elements <- Array new: depth
!!

!Stack methodsFor: 'access'!
empty
    ^ count = 0
!
push: elem
    [count >= elements size]
        ifTrue:  [self error: 'Stack overflow']
        ifFalse: [count <- count + 1.
                  elements at: count put: elem]
!
pop |top|
    [self empty]
        ifTrue:  [self error: 'Stack is empty']
        ifFalse: [top <- elements at: count.
                  count <- count - 1.
                  ^ top]
!!
```