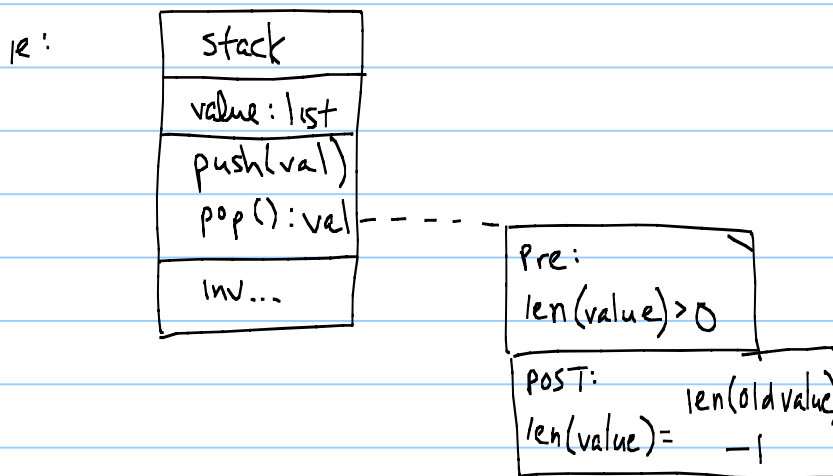


- \* State Space (of a class) - shrunk w/ invariants, types of attributes, ...
  - \* Behaviour: allowed behaviour of a class C is the set of transitions that an obj. of class C is allowed to make between states in its statespace, specified by methods. If some method makes our object be in a state that isn't in our statespace, this is bad design!
- Inheritance  
specialization  
extension

## Method Pre/Post conditions (Design by contract)

Every operation has:

- pre-condition: must be true when operation begins



required: checked by function caller

promise by function. if requirements are satisfied, then this is promised to happen

- post-condition: must be true when operation ends

Design by contract: (between client & operation)

- \* if sender guarantees pre-cond true then operation " post - " "
- \* if sender cannot guarantee pre-cond true,

then... (could STOP, continue, depends on damage that can be caused)

Class inv  $\wedge$  op PRE cond holds

code  $\Downarrow$  sender ensures

(op)

$\Downarrow$  op post cond holds  
still in state  
space