cse 101 1-5-21

- get an account on git.ucsc.edu

  cruzid@ucsc.edu
  username on git.ucsc.edu

  your email address

- Abstract data types (ADT)

  an ADT consists of

  (1) a set of 'mathematical structures' called states: S.

  (2) an assoc. set of operations on ~~states~~.
  - manipulation procedures. (change state)
  - access functions. (return information on a state)

Each ADT instance has a current state, and a history of states.

Example Integer Queue.

States

$S = \{$ all finite sequences of integers, within some bounds: integers x satisfying $L \le x \le U$, and seq length $\le M$. $\}$

Operations

ManiP. Proc.

- Enqueue( ): insert new int at back of sequence.
- Dequeue( ): delete an int from front of seq.

access fcns:
- getFront(): returns front element
- getLength(): returns length of seq.
- isEmpty(): returns True if in empty ~~state~~, false otherwise.

a particular history of ~~states~~:
(client view)

operations	state	return val.
•	( )	-
Enqueue(5)	(5)	-
Enqueue(1)	(5, 1)	-
Enqueue(7)	(5, 1, 7)	-
Dequeue( )	(1, 7)	-
Enqueue(3)	(1, 7, 3)	-
getLength( )	(1, 7, 3)	3
getFront( )	(1, 7, 3)	1
isEmpty( )	(1, 7, 3)	False

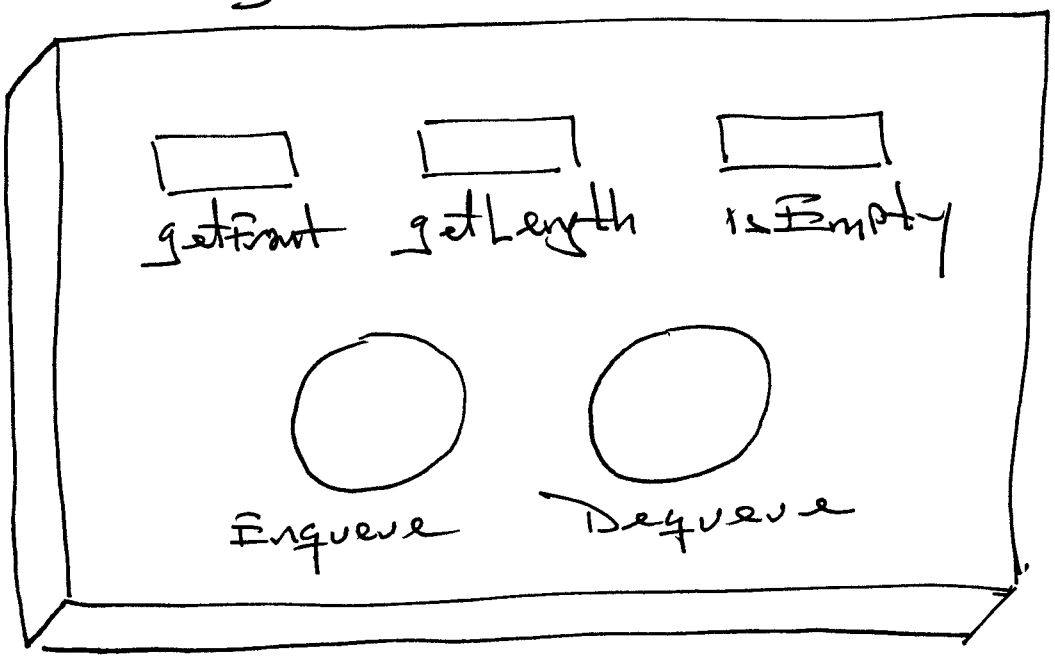
We establish Preconditions for some operations: conditions that must be true when an op. is called,

Policy: if an ADT operation is called on a state that violates its one or more of its preconditions, program end with an error message specifying

- which ADT
- which operation
- which precondition

Think of an ADT instance as being a 'black box'.

Integer queue object



Info-mation hiding: simplify task of client.

modules: