CSc 101-02 1-10-23

Handout on ADTs in C:

Abstract Data Type (ADT)

Defn

An ADT consists of

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

(2) a set of operations on states

- Manipulation Procedures: change state of an ADT instance
- Access functions: return information about a state

Ex. Queue of integers.

states: finite sequences of integers.

	len
e.g. (3, 1, -5, 2, -18)	5
(1, 5, 0)	3
(8)	1
( )	0

operations:

manip. proc.:

Enqueue( )

Dequeue( )

access fcns.:

getFront( )

getLength( )

isEmpty( )

history of states:

Operation	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
⋮	⋮	⋮

Preconditions:

define the set of states to which an op. can be applied.

Policy: if the client violates a precondition, the operation must

(1) Print an error message:

- what ADT
- what operation
- what precondition

Then

(2) exit the program.

All preconditions must be stated in a comment block in terms of access function calls

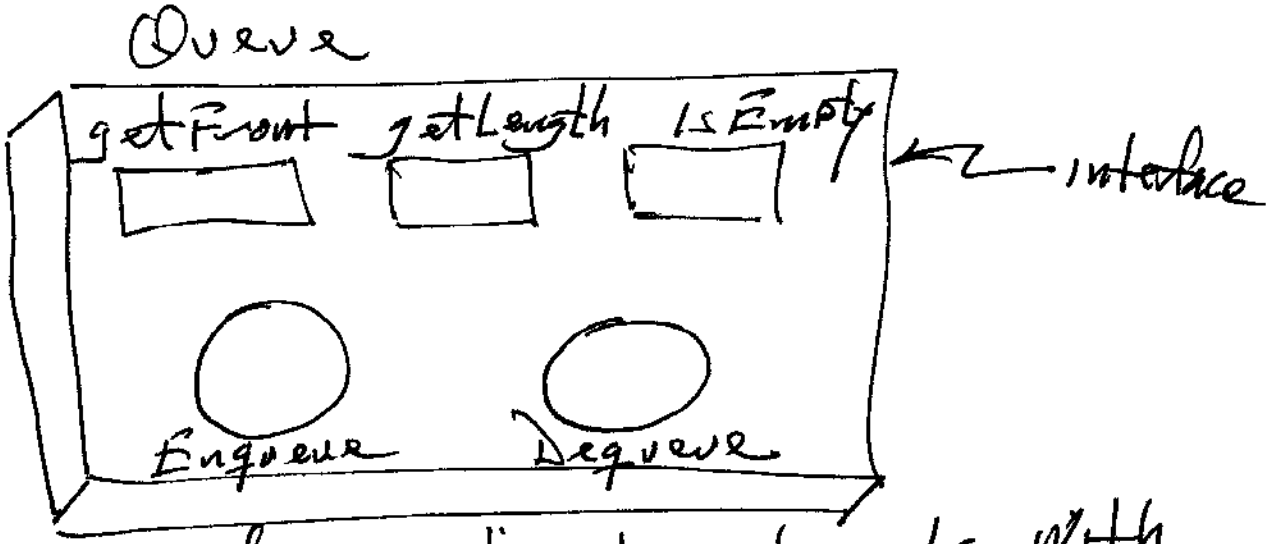
Arguments to ADT ops

in C : Dequeue(Q)

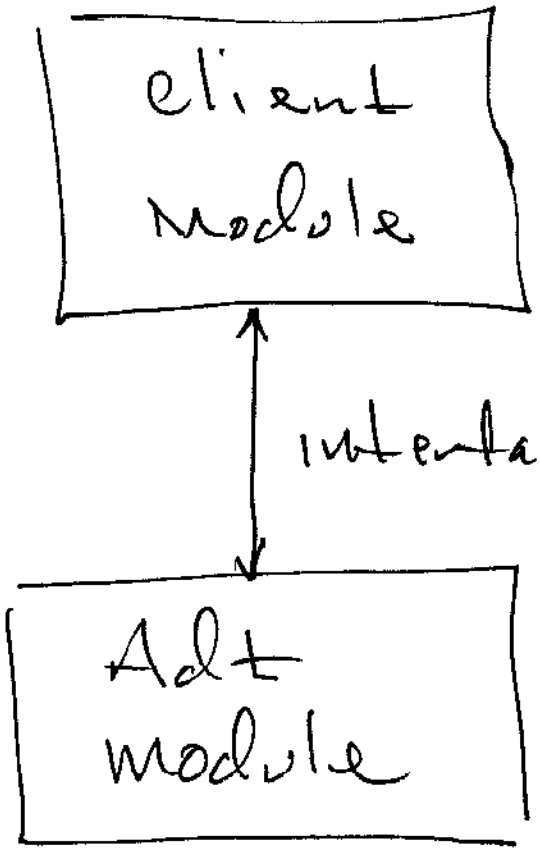
in C++ : Q.Dequeue()

↑ implicit 1st argument

Black Box Picture:



information hiding : client interacts with ADT instance only through the interface.


client
Module
interface
(.h file)
Adt
module
(.c file)