

input variables:	BoundedQueue (int)
state variables:	deQueue

Method	Params
BoundedQueue	int
enQueue	Object
deQueue	state
isEmpty	state
isFull	state

ID	Characteristic
C1	Constructor
C2	If argument is less than 0
C3	Make o the newest element of the queue
C4	If argument is null
C5	Remove and return oldest element of the queue
C6	If queue is empty
C7	If queue is full

Method	Characteristics
BoundedQueue	C1, C2
enQueue	C1, C3, C4, C7
deQueue	C1, C5, C6
isEmpty	C1, C6
isFull	C1, C7

(e) Defin

(a) List all the input variables, including the state variables.

enqueue (Object)	
isEmpty	isFull

(b) Define the characteristics of the input variables. Make sure you cover all the possible values.

Returns	Values	Exception
		IllegalArgumentException
		NullPointerException
		IllegalStateException
Object	Object o	IllegalStateException
boolean	true, false	
boolean	true, false	

(c) Partition the characteristics into blocks. Designate one block in each row.

(d) Define values for each block.

BoundedQueue (int capacity)	enqueue (Object o)	dequeue ()
0	0	0
0	0	
	0	
	0	
		0
		0
	0	

Write a test set that satisfies Base Choice Coverage (BCC). Write your tests with the values from the table above.

Test Requirements	Infeasible TRs	Revised TRs
{TF, FF, TT}	FF	n/a
{TTFF, FTFF, TFFF, TTTF, TTFT}	FTFF, TFFF, TTTF, TTFT	TTTF -> TFTF TTFT -> TTFT
{TTF, FTF, TFF, TTT}	FTF, TFF, TTT	TFF -> TFT
{TF, FT, TT}	FT	n/a
{TF, FT, TT}	FT	n/a

variables.

u cover all input variables.

Ch ID	Characteristic
C1	Constructor
C2	If argument is less than 0
C3	Make o the newest element of the queue
C4	If argument is null
C5	Remove and return oldest element of the queue
C6	If queue is empty
C7	If queue is full

n partition as the "Base" block.

isEmpty()	isFull()
O	O
O	O

from the previous step. Be sure to include the test oracles.

- # TRs
- 2
  - 3
  - 2
  - 2
  - 2

Partition	Base
{true, false}	T
{true, false}	F
{true, false}	T
{true, false}	F
{true, false}	T
{true, false}	F
{true, false}	F