







Assignment 3 – Code Coverage













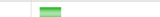


RingBufferExample

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
at.jku.swtesting		100 %		100 %	0	21	0	36	0	13	0	2
Total	0 of 196	100 %	0 of 16	100 %	0	21	0	36	0	13	0	2





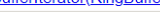

at.jku.swtesting

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
RingBuffer		100 %		100 %	0	15	0	29	0	9	0	1
RingBuffer.RingBufferIterator		100 %		100 %	0	6	0	7	0	4	0	1
Total	0 of 196	100 %	0 of 16	100 %	0	21	0	36	0	13	0	2

RingBuffer

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
enqueue(Object)		100 %		100 %	0	2	0	6	0	1
dequeue()		100 %		100 %	0	2	0	7	0	1
RingBuffer(int)		100 %		100 %	0	2	0	8	0	1
peek()		100 %		100 %	0	2	0	3	0	1
isFull()		100 %		100 %	0	2	0	1	0	1
isEmpty()		100 %		100 %	0	2	0	1	0	1
iterator()		100 %	n/a	n/a	0	1	0	1	0	1
capacity()		100 %	n/a	n/a	0	1	0	1	0	1
size()		100 %	n/a	n/a	0	1	0	1	0	1
Total	0 of 145	100 %	0 of 12	100 %	0	15	0	29	0	9

RingBuffer.RingBufferIterator

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
next()		100 %		100 %	0	2	0	3	0	1
hasNext()		100 %		100 %	0	2	0	1	0	1
RingBuffer.RingBufferIterator(RingBuffer)		100 %	n/a	n/a	0	1	0	2	0	1
remove()		100 %	n/a	n/a	0	1	0	1	0	1
Total	0 of 51	100 %	0 of 4	100 %	0	6	0	7	0	4

Since we already had 100% coverage initially, we didn't have to add any tests in this part of the assignment.