Assignment 3 – Mutation Analysis After

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	100%	36/36	100%	31/31	100%	31/31

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
at.jku.swtesting	1	100%	36/36	100%	31/31	100%	31/31

Package Summary

at.jku.swtesting

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	100%	36/36	100%	31/31	100%	31/31

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
RingBuffer.java	100%	36/36	100%	31/31	100%	31/31

RingBuffer.java

```
1   package at.jku.swtesting;
2
3   import java.util.Iterator;
4   import java.util.NoSuchElementException;
5
6   /**
7    * The RingBuffer class represents a first-in-first-out (FIFO) circular queue of elements.
8    * It has a maximum capacity of elements it can hold. If more elements are added, the
9    * last element will overwrite the first one.
10   *
11   * Originally derived from http://www.cs.princeton.edu/introcs/43stack/RingBuffer.java.html
12   */
13  public class RingBuffer<Item> implements Iterable<Item> {
14
15          private Item[] a;               // queue elements
16          private int N = 0;              // number of elements on queue
17          private int first = 0;  // index of first element of queue
18          private int last = 0;   // index of next available slot
19
20          /**
21           * Creates a new empty ring buffer.
22           * @param capacity number of elements the buffer is able to hold.
23           * @throws IllegalArgumentException if the initial capacity is less than one.
24           */
25          @SuppressWarnings("unchecked")
26          public RingBuffer(int capacity) {
27  2               if (capacity < 1) {
28                          throw new IllegalArgumentException("Initial capacity is less than one");
29                  }
30                  // cast needed since no generic array creation in Java
31                  a = (Item[]) new Object[capacity];
32          }
```

```
33
34          /**
35           * Returns the number of elements the buffer can hold.
36           */
37          public int capacity() {
38 1                return a.length;
39          }
40
41          /**
42           * Returns the number of elements in the buffer.
43           */
44          public int size() {
45 1                return N;
46          }
47
48          /**
49           * Returns true if the buffer contains no elements.
50           */
51          public boolean isEmpty() {
52 2                return N == 0;
53          }
54
55          /**
56           * Returns true if the buffer has reached its capacity, which is the maximum
57           * number of elements it can hold, before overwriting elements.
58           */
59          public boolean isFull() {
60 2                return N == a.length;
61          }
```

```
54
55          /**
56           * Returns true if the buffer has reached its capacity, which is the maximum
57           * number of elements it can hold, before overwriting elements.
58           */
59          public boolean isFull() {
60 2                return N == a.length;
61          }
62
63          /**
64           * Appends the specified element to the end of the buffer. If the buffer has already
65           * reached its capacity, appending overwrites the first element in the buffer.
66           * @param item to be appended to the buffer.
67           */
68          public void enqueue(Item item) {
69                  a[last] = item;
70 2                last = (last + 1) % a.length; // wrap-around
71 2                if (N < a.length) {
72 1                        N++;
73                  } else {
74 2                        first = (first + 1) % a.length;
75                  }
76          }
77
```

```
77
78          /**
79           * Removes the first element from the buffer.
80           * @throws RuntimeException if the buffer is empty.
81           */
82          public Item dequeue() throws RuntimeException {
83 1                if (isEmpty()) {
84                          throw new RuntimeException("Empty ring buffer.");
85                  }
86                  Item item = a[first];
87                  a[first] = null;
88 1                N--;
89 2                first = (first + 1) % a.length; // wrap-around
90 1                return item;
91          }
92
93          /**
94           * Returns the first element from the buffer without removing it.
95           * @throws a RuntimeException if the buffer is empty.
96           */
97          public Item peek() {
98 1                if (isEmpty()) {
99                          throw new RuntimeException("Empty ring buffer.");
100                 }
101 1                return a[first];
102         }

104         /**
105          * Returns an iterator over the elements in the buffer.
106          */
107         public Iterator<Item> iterator() {
108 1                return new RingBufferIterator();
109         }
110
111         private class RingBufferIterator implements Iterator<Item> {
112                 private int i = 0;
113
114                 /** @inheritDoc */
115                 public boolean hasNext() {
116 3                       return i < N;
117                 }
118
119                 /** @inheritDoc */
120                 public void remove() {
121                         // iterator, doesn't implement remove() since it's optional
122                         throw new UnsupportedOperationException();
123                 }
124
125                 /** @inheritDoc */
126                 public Item next() {
127 1                       if (!hasNext())
128                                 throw new NoSuchElementException();
129                         // return a[i++];
130 4                       return a[(first + i++) % a.length];
131                 }
132         }
133
134 }
```

Mutations

27	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
38	1. replaced int return with 0 for at/jku/swtesting/RingBuffer::capacity → KILLED
45	1. replaced int return with 0 for at/jku/swtesting/RingBuffer::size → KILLED
52	1. replaced boolean return with true for at/jku/swtesting/RingBuffer::isEmpty → KILLED 2. negated conditional → KILLED
60	1. negated conditional → KILLED 2. replaced boolean return with true for at/jku/swtesting/RingBuffer::isFull → KILLED
70	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer modulus with multiplication → KILLED
71	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
72	1. Replaced integer addition with subtraction → KILLED
74	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer modulus with multiplication → KILLED
83	1. negated conditional → KILLED
88	1. Replaced integer subtraction with addition → KILLED
89	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer modulus with multiplication → KILLED
90	1. replaced return value with null for at/jku/swtesting/RingBuffer::dequeue → KILLED
98	1. negated conditional → KILLED
101	1. replaced return value with null for at/jku/swtesting/RingBuffer::peek → KILLED
108	1. replaced return value with null for at/jku/swtesting/RingBuffer::iterator → KILLED
116	1. changed conditional boundary → KILLED 2. replaced boolean return with true for at/jku/swtesting/RingBuffer$RingBufferIterator::hasNext → KILLED 3. negated conditional → KILLED
127	1. negated conditional → KILLED
130	1. replaced return value with null for at/jku/swtesting/RingBuffer$RingBufferIterator::next → KILLED 2. Replaced integer modulus with multiplication → KILLED 3. Replaced integer addition with subtraction → KILLED 4. Replaced integer addition with subtraction → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testCreateEmptyBuffer()] (0 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testEmptyPeek()] (0 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testCapacity()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testIsFull()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testEnqueue()] (9 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testDequeue()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testFilledIterator()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testIsEmpty()] (2 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testDequeuedIterator()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testSize()] (0 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testEmptyDequeue()] (2 ms)
- at.jku.swtesting.RingBufferMutationTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferMutationTest]/[method:testCreationCapacity()] (14 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testEmptyIterator()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testPeek()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testIteratorRemove()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testPartiallyFilledIterator()] (1 ms)
- at.jku.swtesting.RingBufferTest.[engine:junit-jupiter]/[class:at.jku.swtesting.RingBufferTest]/[method:testOverwrittenRingBuffer()] (2 ms)