CollectionUtils.java

```
1    package org.usi.sq.util;
2
3    import java.util.ArrayList;
4    import java.util.Collections;
5    import java.util.List;
6
7    public class CollectionUtils {
8
9        private static final double EPSILON = 1e-6;
10
11       /**
12        * Implementation of insertion sort. Insertion sort can be used in cases
13        * where the input list is almost sorted or small. The list implementation
14        * should provide fast indexed access. Due to its quadratic worst-case
15        * runtime complexity, other sorting algorithms should be used in the
16        * general case.
17        *
18        * @param list the list to be sorted in-place
19        * @param <T> the type of the elements in the list (need to be comparable)
20        * @throws IllegalArgumentException if called with null pointer
21        */
22       static <T extends Comparable<T>> void sort(List<T> list){
23 1         if(list == null)
24               throw new IllegalArgumentException("Sort must not be called with " +
25                       "null");
26 3         for(int i = 1; i < list.size(); i++){
27               T key = list.get(i);
28               int j = 0;
29 4             for(j = i-1; j >= 0; j--){
30                   T elem = list.get(j);
31 2                 if(elem.compareTo(key) > 0){ // check if elem > key
32 1                     list.set(j+1, elem);
33                   } else{
34                       break;
35                   }
36               }
37 1             list.set(j+1,key);
38           }
39       }
40   }
```

Mutations

```
23   1. negated conditional → KILLED
     1. changed conditional boundary → KILLED
26   2. Changed increment from 1 to -1 → KILLED
     3. negated conditional → KILLED
     1. changed conditional boundary → KILLED
     2. Changed increment from -1 to 1 → KILLED
29   3. Replaced integer subtraction with addition → KILLED
     4. negated conditional → KILLED
```

```
31  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
32  1. Replaced integer addition with subtraction → KILLED
37  1. Replaced integer addition with subtraction → KILLED
```

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- org.usi.sq.util.CollectionUtilsTest.[engine:junit-jupiter]/[class:org.usi.sq.util.CollectionUtilsTest]/[method:checkSorted()] (6 ms)
- org.usi.sq.util.CollectionUtilsTest.[engine:junit-jupiter]/[class:org.usi.sq.util.CollectionUtilsTest]/[method:checkStability()] (4 ms)
- org.usi.sq.util.CollectionUtilsTest.[engine:junit-jupiter]/[class:org.usi.sq.util.CollectionUtilsTest]/[method:checkConstructor()] (12 ms)

Report generated by PIT 1.5.2