

# Range.java

```

1  /* =====
2  * JFreeChart : a free chart library for the Java(tm) platform
3  * =====
4  *
5  * (C) Copyright 2000-2014, by Object Refinery Limited and Contributors.
6  *
7  * Project Info:  http://www.jfree.org/jfreechart/index.html
8  *
9  * This library is free software; you can redistribute it and/or modify it
10 * under the terms of the GNU Lesser General Public License as published by
11 * the Free Software Foundation; either version 2.1 of the License, or
12 * (at your option) any later version.
13 *
14 * This library is distributed in the hope that it will be useful, but
15 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
16 * or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public
17 * License for more details.
18 *
19 * You should have received a copy of the GNU Lesser General Public
20 * License along with this library; if not, write to the Free Software
21 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
22 * USA.
23 *
24 * [Oracle and Java are registered trademarks of Oracle and/or its affiliates.
25 * Other names may be trademarks of their respective owners.]
26 *
27 * -----
28 * Range.java
29 * -----
30 * (C) Copyright 2002-2014, by Object Refinery Limited and Contributors.
31 *
32 * Original Author:  David Gilbert (for Object Refinery Limited);
33 * Contributor(s):   Chuanhao Chiu;
34 *                   Bill Kelemen;
35 *                   Nicolas Brodu;
36 *                   Sergei Ivanov;
37 *
38 * Changes (from 23-Jun-2001)
39 * -----
40 * 22-Apr-2002 : Version 1, loosely based by code by Bill Kelemen (DG);
41 * 30-Apr-2002 : Added getLength() and getCentralValue() methods.  Changed
42 *               argument check in constructor (DG);
43 * 13-Jun-2002 : Added contains(double) method (DG);
44 * 22-Aug-2002 : Added fix to combine method where both ranges are null, thanks
45 *               to Chuanhao Chiu for reporting and fixing this (DG);
46 * 07-Oct-2002 : Fixed errors reported by Checkstyle (DG);
47 * 26-Mar-2003 : Implemented Serializable (DG);
48 * 14-Aug-2003 : Added equals() method (DG);
49 * 27-Aug-2003 : Added toString() method (BK);

```

```

50  * 11-Sep-2003 : Added Clone Support (NB);
51  * 23-Sep-2003 : Fixed Checkstyle issues (DG);
52  * 25-Sep-2003 : Oops, Range immutable, clone not necessary (NB);
53  * 05-May-2004 : Added constrain() and intersects() methods (DG);
54  * 18-May-2004 : Added expand() method (DG);
55  * ----- JFreeChart 1.0.x -----
56  * 11-Jan-2006 : Added new method expandToInclude(Range, double) (DG);
57  * 18-Dec-2007 : New methods intersects(Range) and scale(...) thanks to Sergei
58  *               Ivanov (DG);
59  * 08-Jan-2012 : New method combineIgnoringNaN() (DG);
60  * 23-Feb-2014 : Added isNaNRange() method (DG);
61  *
62  */
63
64  package org.jfree.data;
65
66  import java.io.Serializable;
67  import org.jfree.chart.util.ParamChecks;
68
69  /**
70   * Represents an immutable range of values.
71   */
72  public strictfp class Range implements Serializable {
73
74      /** For serialization. */
75      private static final long serialVersionUID = -906333695431863380L;
76
77      /** The lower bound of the range. */
78      private double lower;
79
80      /** The upper bound of the range. */
81      private double upper;
82
83      /**
84       * Creates a new range.
85       *
86       * @param lower the lower bound (must be <= upper bound).
87       * @param upper the upper bound (must be >= lower bound).
88       */
89      public Range(double lower, double upper) {
90          19      if (lower > upper) {
91              9      String msg = "Range(double, double): require lower (" + lower
92                  11          + ") <= upper (" + upper + ").";
93              1      throw new IllegalArgumentException(msg);
94          }
95          6      this.lower = lower;
96          6      this.upper = upper;
97      }
98
99      /**
100       * Returns the lower bound for the range.
101       *
102       * @return The lower bound.

```

```

103     */
104     public double getLowerBound() {
105 7         return this.lower;
106     }
107
108     /**
109      * Returns the upper bound for the range.
110      *
111      * @return The upper bound.
112      */
113     public double getUpperBound() {
114 7         return this.upper;
115     }
116
117     /**
118      * Returns the length of the range.
119      *
120      * @return The length.
121      */
122     public double getLength() {
123 19         return this.upper - this.lower;
124     }
125
126     /**
127      * Returns the central value for the range.
128      *
129      * @return The central value.
130      */
131     public double getCentralValue() {
132 47         return this.lower / 2.0 + this.upper / 2.0;
133     }
134
135     /**
136      * Returns true if the range contains the specified value and
137      * false otherwise.
138      *
139      * @param value the value to lookup.
140      *
141      * @return true if the range contains the specified value.
142      */
143     public boolean contains(double value) {
144 53         return (value >= this.lower && value <= this.upper);
145     }
146
147     /**
148      * Returns true if the range intersects with the specified
149      * range, and false otherwise.
150      *
151      * @param b0 the lower bound (should be <= b1).
152      * @param b1 the upper bound (should be >= b0).
153      *
154      * @return A boolean.
155      */

```

```

156     public boolean intersects(double b0, double b1) {
157         19         if (b0 <= this.lower) {
158         34             return (b1 > this.lower);
159         }
160         else {
161         53             return (b0 < this.upper && b1 >= b0);
162         }
163     }
164
165     /**
166      * Returns <code>true</code> if the range intersects with the specified
167      * range, and <code>false</code> otherwise.
168      *
169      * @param range  another range (<code>null</code> not permitted).
170      *
171      * @return A boolean.
172      *
173      * @since 1.0.9
174      */
175     public boolean intersects(Range range) {
176     6         return intersects(range.getLowerBound(), range.getUpperBound());
177     }
178
179     /**
180      * Returns the value within the range that is closest to the specified
181      * value.
182      *
183      * @param value  the value.
184      *
185      * @return The constrained value.
186      */
187     public double constrain(double value) {
188     5         double result = value;
189     14         if (!contains(value)) {
190     19             if (value > this.upper) {
191     5                 result = this.upper;
192             }
193     19             else if (value < this.lower) {
194     5                 result = this.lower;
195             }
196         }
197     7         return result;
198     }
199
200     /**
201      * Creates a new range by combining two existing ranges.
202      * <P>
203      * Note that:
204      * <ul>
205      * <li>either range can be <code>null</code>, in which case the other
206      *     range is returned;</li>
207      * <li>if both ranges are <code>null</code> the return value is
208      *     <code>null</code>.</li>

```

```

209      * </ul>
210      *
211      * @param range1  the first range (<code>>null</code> permitted).
212      * @param range2  the second range (<code>>null</code> permitted).
213      *
214      * @return A new range (possibly <code>>null</code>).
215      */
216      public static Range combine(Range range1, Range range2) {
217 4          if (range1 == null) {
218 2              return range2;
219          }
220 4          if (range2 == null) {
221 2              return range1;
222          }
223 4          double l = Math.min(range1.getLowerBound(), range2.getLowerBound());
224 4          double u = Math.max(range1.getUpperBound(), range2.getUpperBound());
225 13          return new Range(l, u);
226      }
227
228      /**
229      * Returns a new range that spans both <code>range1</code> and
230      * <code>range2</code>. This method has a special handling to ignore
231      * Double.NaN values.
232      *
233      * @param range1  the first range (<code>>null</code> permitted).
234      * @param range2  the second range (<code>>null</code> permitted).
235      *
236      * @return A new range (possibly <code>>null</code>).
237      *
238      * @since 1.0.15
239      */
240      public static Range combineIgnoringNaN(Range range1, Range range2) {
241 4          if (range1 == null) {
242 13              if (range2 != null && range2.isNaNRange()) {
243 1                  return null;
244              }
245 2              return range2;
246          }
247 4          if (range2 == null) {
248 9              if (range1.isNaNRange()) {
249 1                  return null;
250              }
251 2              return range1;
252          }
253 4          double l = min(range1.getLowerBound(), range2.getLowerBound());
254 4          double u = max(range1.getUpperBound(), range2.getUpperBound());
255 28          if (Double.isNaN(l) && Double.isNaN(u)) {
256 1              return null;
257          }
258 13          return new Range(l, u);
259      }
260
261      /**

```

```

262     * Returns the minimum value.  If either value is NaN, the other value is
263     * returned.  If both are NaN, NaN is returned.
264     *
265     * @param d1  value 1.
266     * @param d2  value 2.
267     *
268     * @return The minimum of the two values.
269     */
270     private static double min(double d1, double d2) {
271 14         if (Double.isNaN(d1)) {
272 7             return d2;
273         }
274 14         if (Double.isNaN(d2)) {
275 7             return d1;
276         }
277 14         return Math.min(d1, d2);
278     }
279
280     private static double max(double d1, double d2) {
281 14         if (Double.isNaN(d1)) {
282 7             return d2;
283         }
284 14         if (Double.isNaN(d2)) {
285 7             return d1;
286         }
287 14         return Math.max(d1, d2);
288     }
289
290     /**
291     * Returns a range that includes all the values in the specified
292     * <code>range</code> AND the specified <code>value</code>.
293     *
294     * @param range  the range (<code>null</code> permitted).
295     * @param value  the value that must be included.
296     *
297     * @return A range.
298     *
299     * @since 1.0.1
300     */
301     public static Range expandToInclude(Range range, double value) {
302 4         if (range == null) {
303 13             return new Range(value, value);
304         }
305 15         if (value < range.getLowerBound()) {
306 9             return new Range(value, range.getUpperBound());
307         }
308 15         else if (value > range.getUpperBound()) {
309 9             return new Range(range.getLowerBound(), value);
310         }
311         else {
312 2             return range;
313         }
314     }

```

```

315
316 /**
317  * Creates a new range by adding margins to an existing range.
318  *
319  * @param range the range (<code>null</code> not permitted).
320  * @param lowerMargin the lower margin (expressed as a percentage of the
321  * range length).
322  * @param upperMargin the upper margin (expressed as a percentage of the
323  * range length).
324  *
325  * @return The expanded range.
326  */
327 public static Range expand(Range range,
328                           double lowerMargin, double upperMargin) {
329 1     ParamChecks.nullNotPermitted(range, "range");
330 1     double length = range.getLength();
331 25     double lower = range.getLowerBound() - length * lowerMargin;
332 25     double upper = range.getUpperBound() + length * upperMargin;
333 19     if (lower > upper) {
334 45         lower = lower / 2.0 + upper / 2.0;
335 5         upper = lower;
336     }
337 13     return new Range(lower, upper);
338 }
339
340 /**
341  * Shifts the range by the specified amount.
342  *
343  * @param base the base range (<code>null</code> not permitted).
344  * @param delta the shift amount.
345  *
346  * @return A new range.
347  */
348 public static Range shift(Range base, double delta) {
349 14     return shift(base, delta, false);
350 }
351
352 /**
353  * Shifts the range by the specified amount.
354  *
355  * @param base the base range (<code>null</code> not permitted).
356  * @param delta the shift amount.
357  * @param allowZeroCrossing a flag that determines whether or not the
358  * bounds of the range are allowed to cross
359  * zero after adjustment.
360  *
361  * @return A new range.
362  */
363 public static Range shift(Range base, double delta,
364                           boolean allowZeroCrossing) {
365 1     ParamChecks.nullNotPermitted(base, "base");
366 13     if (allowZeroCrossing) {
367 16         return new Range(base.getLowerBound() + delta,

```

```

368 13         base.getUpperBound() + delta);
369     }
370     else {
371 6         return new Range(shiftWithNoZeroCrossing(base.getLowerBound(),
372 8             delta), shiftWithNoZeroCrossing(base.getUpperBound(),
373 5             delta));
374     }
375 }
376
377 /**
378  * Returns the given value adjusted by delta but
379  * with a check to prevent the result from crossing 0.0.
380  *
381  * @param value the value.
382  * @param delta the adjustment.
383  *
384  * @return The adjusted value.
385  */
386 private static double shiftWithNoZeroCrossing(double value, double delta) {
387 19     if (value > 0.0) {
388 26         return Math.max(value + delta, 0.0);
389     }
390 19     else if (value < 0.0) {
391 26         return Math.min(value + delta, 0.0);
392     }
393     else {
394 19         return value + delta;
395     }
396 }
397
398 /**
399  * Scales the range by the specified factor.
400  *
401  * @param base the base range (null not permitted).
402  * @param factor the scaling factor (must be non-negative).
403  *
404  * @return A new range.
405  *
406  * @since 1.0.9
407  */
408 public static Range scale(Range base, double factor) {
409 1     ParamChecks.nullNotPermitted(base, "base");
410 19     if (factor < 0) {
411 1         throw new IllegalArgumentException("Negative 'factor' argument.");
412     }
413 16     return new Range(base.getLowerBound() * factor,
414 13         base.getUpperBound() * factor);
415 }
416
417 /**
418  * Tests this object for equality with an arbitrary object.
419  *
420  * @param obj the object to test against (null permitted).

```



```

421     *
422     * @return A boolean.
423     */
424     @Override
425     public boolean equals(Object obj) {
426 8         if (!(obj instanceof Range)) {
427 7             return false;
428         }
429         Range range = (Range) obj;
430 18         if (!(this.lower == range.lower)) {
431 7             return false;
432         }
433 18         if (!(this.upper == range.upper)) {
434 7             return false;
435         }
436 8         return true;
437     }
438
439     /**
440     * Returns <code>true</code> if both the lower and upper bounds are
441     * <code>Double.NaN</code>, and <code>false</code> otherwise.
442     *
443     * @return A boolean.
444     *
445     * @since 1.0.18
446     */
447     public boolean isNaNRange() {
448 43         return Double.isNaN(this.lower) && Double.isNaN(this.upper);
449     }
450
451     /**
452     * Returns a hash code.
453     *
454     * @return A hash code.
455     */
456     @Override
457     public int hashCode() {
458         int result;
459         long temp;
460 6         temp = Double.doubleToLongBits(this.lower);
461 19         result = (int) (temp ^ (temp >> 32));
462 6         temp = Double.doubleToLongBits(this.upper);
463 45         result = 29 * result + (int) (temp ^ (temp >> 32));
464 7         return result;
465     }
466
467     /**
468     * Returns a string representation of this Range.
469     *
470     * @return A String "Range[lower,upper]" where lower=lower range and
471     *         upper=upper range.
472     */
473     @Override

```

```

474     public String toString() {
475 22         return ("Range[" + this.lower + "," + this.upper + "]");
476     }
477
478 }

```

## Mutations

1. changed conditional boundary → KILLED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED
5. Negated double local variable number 1 → KILLED
6. Negated double local variable number 3 → KILLED
7. Less or equal to less than → KILLED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
- [90](#) 10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Incremented (a++) double local variable number 3 → KILLED
14. Decrementd (a--) double local variable number 1 → KILLED
15. Decrementd (a--) double local variable number 3 → KILLED
16. Incremented (++a) double local variable number 1 → KILLED
17. Incremented (++a) double local variable number 3 → KILLED
18. Decrementd (--a) double local variable number 1 → KILLED
19. Decrementd (--a) double local variable number 3 → KILLED
1. removed call to java/lang/StringBuilder::<init> → KILLED
2. removed call to java/lang/StringBuilder::append → KILLED
3. removed call to java/lang/StringBuilder::toString → SURVIVED
4. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
- [91](#) 5. Negated double local variable number 1 → SURVIVED
6. Incremented (a++) double local variable number 1 → SURVIVED
7. Decrementd (a--) double local variable number 1 → SURVIVED
8. Incremented (++a) double local variable number 1 → SURVIVED
9. Decrementd (--a) double local variable number 1 → SURVIVED
1. removed call to java/lang/StringBuilder::append → KILLED
2. removed call to java/lang/StringBuilder::append → KILLED
3. removed call to java/lang/StringBuilder::append → KILLED
4. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
5. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
- [92](#) 6. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
7. Negated double local variable number 3 → SURVIVED
8. Incremented (a++) double local variable number 3 → SURVIVED
9. Decrementd (a--) double local variable number 3 → SURVIVED
10. Incremented (++a) double local variable number 3 → SURVIVED
11. Decrementd (--a) double local variable number 3 → SURVIVED
- [93](#) 1. removed call to java/lang/IllegalArgumentException::<init> → KILLED
1. Removed assignment to member variable lower → KILLED
2. Negated double local variable number 1 → KILLED
- [95](#) 3. Incremented (a++) double local variable number 1 → SURVIVED
4. Decrementd (a--) double local variable number 1 → SURVIVED
5. Incremented (++a) double local variable number 1 → KILLED
6. Decrementd (--a) double local variable number 1 → KILLED
1. Removed assignment to member variable upper → KILLED
2. Negated double local variable number 3 → KILLED
- [96](#) 3. Incremented (a++) double local variable number 3 → SURVIVED
4. Decrementd (a--) double local variable number 3 → SURVIVED
5. Incremented (++a) double local variable number 3 → KILLED
6. Decrementd (--a) double local variable number 3 → KILLED
- [105](#) 1. replaced double return with 0.0d for org/jfree/data/Range::getLowerBound → KILLED

2. replaced return of double value with  $-(x + 1)$  for  
 org/jfree/data/Range::getLowerBound → KILLED

3. Negated double field lower → KILLED

4. Incremented (a++) double field lower → KILLED

5. Decrementd (a--) double field lower → KILLED

6. Incremented (++a) double field lower → KILLED

7. Decrementd (--a) double fieldlower → KILLED

1. replaced double return with 0.0d for org/jfree/data/Range::getUpperBound → KILLED

2. replaced return of double value with  $-(x + 1)$  for  
 org/jfree/data/Range::getUpperBound → KILLED

[114](#) 3. Negated double field upper → KILLED

4. Incremented (a++) double field upper → KILLED

5. Decrementd (a--) double field upper → KILLED

6. Incremented (++a) double field upper → KILLED

7. Decrementd (--a) double fieldupper → KILLED

1. Replaced double subtraction with addition → KILLED

2. replaced double return with 0.0d for org/jfree/data/Range::getLength → KILLED

3. replaced return of double value with  $-(x + 1)$  for  
 org/jfree/data/Range::getLength → KILLED

4. Negated double field upper → KILLED

5. Negated double field lower → KILLED

6. Replaced double operation with first member → KILLED

7. Replaced double operation by second member → KILLED

8. Replaced double subtraction with addition → KILLED

[123](#) 9. Replaced double subtraction with multiplication → KILLED

10. Replaced double subtraction with division → KILLED

11. Replaced double subtraction with modulus → KILLED

12. Incremented (a++) double field upper → KILLED

13. Incremented (a++) double field lower → KILLED

14. Decrementd (a--) double field upper → KILLED

15. Decrementd (a--) double field lower → KILLED

16. Incremented (++a) double field upper → KILLED

17. Incremented (++a) double field lower → KILLED

18. Decrementd (--a) double fieldupper → KILLED

19. Decrementd (--a) double fieldlower → KILLED

[132](#) 1. Substituted 2.0 with 1.0 → KILLED

2. Substituted 2.0 with 1.0 → KILLED

3. Replaced double division with multiplication → KILLED

4. Replaced double division with multiplication → KILLED

5. Replaced double addition with subtraction → KILLED

6. replaced double return with 0.0d for org/jfree/data/Range::getCentralValue → SURVIVED

7. replaced return of double value with  $-(x + 1)$  for  
 org/jfree/data/Range::getCentralValue → KILLED

8. Negated double field lower → KILLED

9. Negated double field upper → KILLED

10. Replaced double operation with first member → KILLED

11. Replaced double operation with first member → KILLED

12. Replaced double operation with first member → KILLED

13. Replaced double operation by second member → KILLED

14. Replaced double operation by second member → KILLED

15. Replaced double operation by second member → KILLED

16. Replaced double division with multiplication → KILLED

17. Replaced double division with multiplication → KILLED

18. Replaced double addition with subtraction → KILLED

19. Replaced double division with modulus → KILLED

20. Replaced double division with modulus → KILLED

21. Replaced double addition with multiplication → KILLED

22. Replaced double division with addition → KILLED

23. Replaced double division with addition → KILLED

24. Replaced double addition with division → KILLED

25. Replaced double division with subtraction → KILLED

26. Replaced double division with subtraction → KILLED

27. Replaced double addition with modulus → SURVIVED

28. Substituted 2.0 with 1.0 → KILLED  
 29. Substituted 2.0 with 1.0 → KILLED  
 30. Substituted 2.0 with 0.0 → KILLED  
 31. Substituted 2.0 with 0.0 → KILLED  
 32. Substituted 2.0 with -1.0 → KILLED  
 33. Substituted 2.0 with -1.0 → KILLED  
 34. Substituted 2.0 with -2.0 → KILLED  
 35. Substituted 2.0 with -2.0 → KILLED  
 36. Substituted 2.0 with 3.0 → KILLED  
 37. Substituted 2.0 with 3.0 → KILLED  
 38. Substituted 2.0 with 1.0 → KILLED  
 39. Substituted 2.0 with 1.0 → KILLED  
 40. Incremented (a++) double field lower → SURVIVED  
 41. Incremented (a++) double field upper → SURVIVED  
 42. Decrementd (a--) double field lower → SURVIVED  
 43. Decrementd (a--) double field upper → SURVIVED  
 44. Incremented (++a) double field lower → KILLED  
 45. Incremented (++a) double field upper → KILLED  
 46. Decrementd (--a) double fieldlower → KILLED  
 47. Decrementd (--a) double fieldupper → KILLED

[144](#) 1. replaced boolean return with false for org/jfree/data/Range::contains → SURVIVED  
 2. replaced boolean return with true for org/jfree/data/Range::contains → KILLED  
 3. changed conditional boundary → SURVIVED  
 4. changed conditional boundary → SURVIVED  
 5. Substituted 1 with 0 → SURVIVED  
 6. Substituted 0 with 1 → KILLED  
 7. negated conditional → KILLED  
 8. negated conditional → KILLED  
 9. removed conditional - replaced comparison check with false → SURVIVED  
 10. removed conditional - replaced comparison check with false → SURVIVED  
 11. removed conditional - replaced comparison check with true → KILLED  
 12. removed conditional - replaced comparison check with true → KILLED  
 13. replaced return of integer sized value with (x == 0 ? 1 : 0) → SURVIVED  
 14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 15. Negated double local variable number 1 → KILLED  
 16. Negated double field lower → SURVIVED  
 17. Negated double local variable number 1 → KILLED  
 18. Negated double field upper → SURVIVED  
 19. Substituted 0 with 1 → KILLED  
 20. Substituted 1 with 0 → SURVIVED  
 21. Substituted 1 with -1 → SURVIVED  
 22. Substituted 0 with -1 → KILLED  
 23. Substituted 1 with -1 → SURVIVED  
 24. Substituted 1 with 2 → SURVIVED  
 25. Substituted 0 with 1 → KILLED  
 26. Substituted 1 with 0 → SURVIVED  
 27. Substituted 0 with -1 → KILLED  
 28. Less than to less or equal → SURVIVED  
 29. greater than to less than → KILLED  
 30. Less than to greater than → KILLED  
 31. greater than to less or equal → KILLED  
 32. Less than to greater or equal → KILLED  
 33. greater than to greater or equal → SURVIVED  
 34. Less than to equal → KILLED  
 35. greater than to equal → KILLED  
 36. Less than to not equal → SURVIVED  
 37. greater than to not equal → SURVIVED  
 38. Incremented (a++) double local variable number 1 → SURVIVED  
 39. Incremented (a++) double field lower → KILLED  
 40. Incremented (a++) double local variable number 1 → SURVIVED  
 41. Incremented (a++) double field upper → KILLED  
 42. Decrementd (a--) double local variable number 1 → KILLED  
 43. Decrementd (a--) double field lower → KILLED  
 44. Decrementd (a--) double local variable number 1 → SURVIVED

45. Decrementd (a--) double field upper → KILLED  
 46. Incremented (++a) double local variable number 1 → KILLED  
 47. Incremented (++a) double field lower → KILLED  
 48. Incremented (++a) double local variable number 1 → SURVIVED  
 49. Incremented (++a) double field upper → KILLED  
 50. Decrementd (--a) double local variable number 1 → KILLED  
 51. Decrementd (--a) double fieldlower → KILLED  
 52. Decrementd (--a) double local variable number 1 → KILLED  
 53. Decrementd (--a) double fieldupper → KILLED  
 1. changed conditional boundary → KILLED  
 2. negated conditional → KILLED  
 3. removed conditional - replaced comparison check with false → KILLED  
 4. removed conditional - replaced comparison check with true → KILLED  
 5. Negated double local variable number 1 → KILLED  
 6. Negated double field lower → KILLED  
 7. greater than to less than → KILLED  
 8. greater than to less or equal → KILLED  
 9. greater than to greater or equal → KILLED  
 157 10. greater than to equal → KILLED  
 11. greater than to not equal → KILLED  
 12. Incremented (a++) double local variable number 1 → KILLED  
 13. Incremented (a++) double field lower → KILLED  
 14. Decrementd (a--) double local variable number 1 → KILLED  
 15. Decrementd (a--) double field lower → KILLED  
 16. Incremented (++a) double local variable number 1 → KILLED  
 17. Incremented (++a) double field lower → KILLED  
 18. Decrementd (--a) double local variable number 1 → KILLED  
 19. Decrementd (--a) double fieldlower → KILLED  
 1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED  
 2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED  
 3. changed conditional boundary → KILLED  
 4. Substituted 1 with 0 → KILLED  
 5. Substituted 0 with 1 → KILLED  
 6. negated conditional → KILLED  
 7. removed conditional - replaced comparison check with false → KILLED  
 8. removed conditional - replaced comparison check with true → KILLED  
 9. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 10. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 11. Negated double local variable number 3 → KILLED  
 12. Negated double field lower → KILLED  
 13. Substituted 0 with 1 → KILLED  
 14. Substituted 1 with 0 → KILLED  
 15. Substituted 1 with -1 → SURVIVED  
 158 16. Substituted 0 with -1 → KILLED  
 17. Substituted 1 with -1 → SURVIVED  
 18. Substituted 1 with 2 → KILLED  
 19. Substituted 0 with 1 → KILLED  
 20. Substituted 1 with 0 → KILLED  
 21. Substituted 0 with -1 → KILLED  
 22. Less or equal to less than → KILLED  
 23. Less or equal to greater than → KILLED  
 24. Less or equal to greater or equal → KILLED  
 25. Less or equal to equal → KILLED  
 26. Less or equal to not equal → KILLED  
 27. Incremented (a++) double local variable number 3 → SURVIVED  
 28. Incremented (a++) double field lower → KILLED  
 29. Decrementd (a--) double local variable number 3 → SURVIVED  
 30. Decrementd (a--) double field lower → KILLED  
 31. Incremented (++a) double local variable number 3 → KILLED  
 32. Incremented (++a) double field lower → KILLED  
 33. Decrementd (--a) double local variable number 3 → KILLED  
 34. Decrementd (--a) double fieldlower → KILLED  
 161 1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED

2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED

3. changed conditional boundary → KILLED

4. changed conditional boundary → KILLED

5. Substituted 1 with 0 → KILLED

6. Substituted 0 with 1 → KILLED

7. negated conditional → KILLED

8. negated conditional → KILLED

9. removed conditional - replaced comparison check with false → KILLED

10. removed conditional - replaced comparison check with false → KILLED

11. removed conditional - replaced comparison check with true → KILLED

12. removed conditional - replaced comparison check with true → KILLED

13. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED

14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED

15. Negated double local variable number 1 → KILLED

16. Negated double field upper → KILLED

17. Negated double local variable number 3 → KILLED

18. Negated double local variable number 1 → KILLED

19. Substituted 0 with 1 → KILLED

20. Substituted 1 with 0 → KILLED

21. Substituted 1 with -1 → SURVIVED

22. Substituted 0 with -1 → KILLED

23. Substituted 1 with -1 → SURVIVED

24. Substituted 1 with 2 → KILLED

25. Substituted 0 with 1 → KILLED

26. Substituted 1 with 0 → KILLED

27. Substituted 0 with -1 → KILLED

28. greater or equal to less than → KILLED

29. Less than to less or equal → KILLED

30. greater or equal to less or equal → KILLED

31. Less than to greater than → KILLED

32. greater or equal to greater than → KILLED

33. Less than to greater or equal → KILLED

34. greater or equal to equal → KILLED

35. Less than to equal → KILLED

36. greater or equal to not equal → KILLED

37. Less than to not equal → KILLED

38. Incremented (a++) double local variable number 1 → KILLED

39. Incremented (a++) double field upper → SURVIVED

40. Incremented (a++) double local variable number 3 → SURVIVED

41. Incremented (a++) double local variable number 1 → SURVIVED

42. Decrementd (a--) double local variable number 1 → KILLED

43. Decrementd (a--) double field upper → SURVIVED

44. Decrementd (a--) double local variable number 3 → SURVIVED

45. Decrementd (a--) double local variable number 1 → SURVIVED

46. Incremented (++) double local variable number 1 → KILLED

47. Incremented (++) double field upper → KILLED

48. Incremented (++) double local variable number 3 → KILLED

49. Incremented (++) double local variable number 1 → KILLED

50. Decrementd (--a) double local variable number 1 → KILLED

51. Decrementd (--a) double fieldupper → KILLED

52. Decrementd (--a) double local variable number 3 → KILLED

53. Decrementd (--a) double local variable number 1 → KILLED

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED

2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED

3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED

4. removed call to org/jfree/data/Range::getUpperBound → KILLED

5. removed call to org/jfree/data/Range::intersects → KILLED

6. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED

1. Negated double local variable number 1 → KILLED

2. Incremented (a++) double local variable number 1 → KILLED

3. Decrementd (a--) double local variable number 1 → KILLED

4. Incremented (++) double local variable number 1 → KILLED

5. Decrementd (--a) double local variable number 1 → KILLED

176

188



- [189](#)
1. negated conditional → KILLED
  2. removed call to org/jfree/data/Range::contains → SURVIVED
  3. removed conditional - replaced equality check with false → KILLED
  4. removed conditional - replaced equality check with true → SURVIVED
  5. Negated double local variable number 1 → KILLED
  6. not equal to less than → SURVIVED
  7. not equal to less or equal → KILLED
  8. not equal to greater than → SURVIVED
  9. not equal to greater or equal → KILLED
  10. not equal to equal → KILLED
  11. Incremented (a++) double local variable number 1 → KILLED
  12. Decrementd (a--) double local variable number 1 → KILLED
  13. Incremented (++a) double local variable number 1 → KILLED
  14. Decrementd (--a) double local variable number 1 → KILLED
- [190](#)
1. changed conditional boundary → SURVIVED
  2. negated conditional → KILLED
  3. removed conditional - replaced comparison check with false → KILLED
  4. removed conditional - replaced comparison check with true → KILLED
  5. Negated double local variable number 1 → KILLED
  6. Negated double field upper → KILLED
  7. Less or equal to less than → SURVIVED
  8. Less or equal to greater than → KILLED
  9. Less or equal to greater or equal → KILLED
  10. Less or equal to equal → KILLED
  11. Less or equal to not equal → KILLED
  12. Incremented (a++) double local variable number 1 → KILLED
  13. Incremented (a++) double field upper → KILLED
  14. Decrementd (a--) double local variable number 1 → SURVIVED
  15. Decrementd (a--) double field upper → KILLED
  16. Incremented (++a) double local variable number 1 → KILLED
  17. Incremented (++a) double field upper → KILLED
  18. Decrementd (--a) double local variable number 1 → KILLED
  19. Decrementd (--a) double fieldupper → KILLED
- [191](#)
1. Negated double field upper → KILLED
  2. Incremented (a++) double field upper → SURVIVED
  3. Decrementd (a--) double field upper → SURVIVED
  4. Incremented (++a) double field upper → KILLED
  5. Decrementd (--a) double fieldupper → KILLED
  1. changed conditional boundary → SURVIVED
  2. negated conditional → KILLED
  3. removed conditional - replaced comparison check with false → KILLED
  4. removed conditional - replaced comparison check with true → SURVIVED
  5. Negated double local variable number 1 → KILLED
  6. Negated double field lower → SURVIVED
  7. greater or equal to less than → KILLED
  8. greater or equal to less or equal → KILLED
  9. greater or equal to greater than → SURVIVED
- [193](#)
10. greater or equal to equal → SURVIVED
  11. greater or equal to not equal → KILLED
  12. Incremented (a++) double local variable number 1 → SURVIVED
  13. Incremented (a++) double field lower → KILLED
  14. Decrementd (a--) double local variable number 1 → SURVIVED
  15. Decrementd (a--) double field lower → KILLED
  16. Incremented (++a) double local variable number 1 → KILLED
  17. Incremented (++a) double field lower → KILLED
  18. Decrementd (--a) double local variable number 1 → SURVIVED
  19. Decrementd (--a) double fieldlower → KILLED
- [194](#)
1. Negated double field lower → KILLED
  2. Incremented (a++) double field lower → SURVIVED
  3. Decrementd (a--) double field lower → SURVIVED
  4. Incremented (++a) double field lower → KILLED
  5. Decrementd (--a) double fieldlower → KILLED
- [197](#)
1. replaced double return with 0.0d for org/jfree/data/Range::constrain → KILLED

- 2. replaced return of double value with  $-(x + 1)$  for `org/jfree/data/Range::constrain` → KILLED
- 3. Negated double local variable number 3 → KILLED
- 4. Incremented (`a++`) double local variable number 3 → SURVIVED
- 5. Decrementd (`a--`) double local variable number 3 → SURVIVED
- 6. Incremented (`++a`) double local variable number 3 → KILLED
- 7. Decrementd (`--a`) double local variable number 3 → KILLED
- 1. negated conditional → KILLED
- [217](#) 2. removed conditional - replaced equality check with false → KILLED
- 3. removed conditional - replaced equality check with true → KILLED
- 4. not equal to equal → KILLED
- 1. replaced return value with null for `org/jfree/data/Range::combine` → KILLED
- [218](#) 2. mutated return of Object value for `org/jfree/data/Range::combine` to ( if `(x != null)` null else throw new RuntimeException ) → KILLED
- 1. negated conditional → KILLED
- [220](#) 2. removed conditional - replaced equality check with false → KILLED
- 3. removed conditional - replaced equality check with true → KILLED
- 4. not equal to equal → KILLED
- 1. replaced return value with null for `org/jfree/data/Range::combine` → KILLED
- [221](#) 2. mutated return of Object value for `org/jfree/data/Range::combine` to ( if `(x != null)` null else throw new RuntimeException ) → KILLED
- 1. replaced call to `java/lang/Math::min` with argument → KILLED
- [223](#) 2. removed call to `org/jfree/data/Range::getLowerBound` → KILLED
- 3. removed call to `org/jfree/data/Range::getLowerBound` → KILLED
- 4. removed call to `java/lang/Math::min` → KILLED
- 1. replaced call to `java/lang/Math::max` with argument → KILLED
- [224](#) 2. removed call to `org/jfree/data/Range::getUpperBound` → KILLED
- 3. removed call to `org/jfree/data/Range::getUpperBound` → KILLED
- 4. removed call to `java/lang/Math::max` → KILLED
- 1. removed call to `org/jfree/data/Range::<init>` → KILLED
- 2. replaced return value with null for `org/jfree/data/Range::combine` → KILLED
- 3. mutated return of Object value for `org/jfree/data/Range::combine` to ( if `(x != null)` null else throw new RuntimeException ) → KILLED
- 4. Negated double local variable number 2 → KILLED
- 5. Negated double local variable number 4 → KILLED
- [225](#) 6. Incremented (`a++`) double local variable number 2 → SURVIVED
- 7. Incremented (`a++`) double local variable number 4 → SURVIVED
- 8. Decrementd (`a--`) double local variable number 2 → SURVIVED
- 9. Decrementd (`a--`) double local variable number 4 → SURVIVED
- 10. Incremented (`++a`) double local variable number 2 → KILLED
- 11. Incremented (`++a`) double local variable number 4 → KILLED
- 12. Decrementd (`--a`) double local variable number 2 → KILLED
- 13. Decrementd (`--a`) double local variable number 4 → KILLED
- 1. negated conditional → KILLED
- [241](#) 2. removed conditional - replaced equality check with false → KILLED
- 3. removed conditional - replaced equality check with true → KILLED
- 4. not equal to equal → KILLED
- 1. negated conditional → KILLED
- 2. negated conditional → KILLED
- 3. removed call to `org/jfree/data/Range::isNaNRange` → KILLED
- 4. removed conditional - replaced equality check with false → KILLED
- 5. removed conditional - replaced equality check with false → KILLED
- 6. removed conditional - replaced equality check with true → KILLED
- [242](#) 7. removed conditional - replaced equality check with true → KILLED
- 8. equal to less than → KILLED
- 9. equal to less or equal → SURVIVED
- 10. equal to greater than → KILLED
- 11. equal to greater or equal → KILLED
- 12. equal to not equal → KILLED
- 13. equal to not equal → KILLED
- 1. mutated return of Object value for `org/jfree/data/Range::combineIgnoringNaN` to ( if `(x != null)` null else throw new RuntimeException ) → KILLED
- [243](#)
- [245](#) 1. replaced return value with null for `org/jfree/data/Range::combineIgnoringNaN` → KILLED



2. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED

[247](#) 1. negated conditional → KILLED  
 2. removed conditional - replaced equality check with false → KILLED  
 3. removed conditional - replaced equality check with true → KILLED  
 4. not equal to equal → KILLED  
 1. negated conditional → KILLED  
 2. removed call to org/jfree/data/Range::isNaNRange → KILLED  
 3. removed conditional - replaced equality check with false → KILLED  
 4. removed conditional - replaced equality check with true → KILLED

[248](#) 5. equal to less than → KILLED  
 6. equal to less or equal → SURVIVED  
 7. equal to greater than → KILLED  
 8. equal to greater or equal → KILLED  
 9. equal to not equal → KILLED

[249](#) 1. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED

[251](#) 1. replaced return value with null for org/jfree/data/Range::combineIgnoringNaN → KILLED  
 2. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED

[253](#) 1. replaced call to org/jfree/data/Range::min with argument → KILLED  
 2. removed call to org/jfree/data/Range::getLowerBound → KILLED  
 3. removed call to org/jfree/data/Range::getLowerBound → KILLED  
 4. removed call to org/jfree/data/Range::min → KILLED

[254](#) 1. replaced call to org/jfree/data/Range::max with argument → KILLED  
 2. removed call to org/jfree/data/Range::getUpperBound → KILLED  
 3. removed call to org/jfree/data/Range::getUpperBound → KILLED  
 4. removed call to org/jfree/data/Range::max → KILLED

1. negated conditional → KILLED  
 2. negated conditional → KILLED  
 3. removed call to java/lang/Double::isNaN → KILLED  
 4. removed call to java/lang/Double::isNaN → KILLED  
 5. removed conditional - replaced equality check with false → KILLED  
 6. removed conditional - replaced equality check with false → KILLED  
 7. removed conditional - replaced equality check with true → KILLED  
 8. removed conditional - replaced equality check with true → KILLED  
 9. Negated double local variable number 2 → SURVIVED  
 10. Negated double local variable number 4 → SURVIVED  
 11. equal to less than → KILLED  
 12. equal to less than → KILLED  
 13. equal to less or equal → SURVIVED  
 14. equal to less or equal → SURVIVED

[255](#) 15. equal to greater than → KILLED  
 16. equal to greater than → KILLED  
 17. equal to greater or equal → KILLED  
 18. equal to greater or equal → KILLED  
 19. equal to not equal → KILLED  
 20. equal to not equal → KILLED  
 21. Incremented (a++) double local variable number 2 → KILLED  
 22. Incremented (a++) double local variable number 4 → KILLED  
 23. Decrementd (a--) double local variable number 2 → KILLED  
 24. Decrementd (a--) double local variable number 4 → KILLED  
 25. Incremented (a++) double local variable number 2 → KILLED  
 26. Incremented (a++) double local variable number 4 → KILLED  
 27. Decrementd (a--) double local variable number 2 → KILLED  
 28. Decrementd (a--) double local variable number 4 → KILLED

[256](#) 1. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED

[258](#) 1. removed call to org/jfree/data/Range::<init> → KILLED  
 2. replaced return value with null for org/jfree/data/Range::combineIgnoringNaN → KILLED  
 3. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED

4. Negated double local variable number 2 → KILLED  
 5. Negated double local variable number 4 → KILLED  
 6. Incremented (a++) double local variable number 2 → SURVIVED  
 7. Incremented (a++) double local variable number 4 → SURVIVED  
 8. Decrementd (a--) double local variable number 2 → SURVIVED  
 9. Decrementd (a--) double local variable number 4 → SURVIVED  
 10. Incremented (++a) double local variable number 2 → KILLED  
 11. Incremented (++a) double local variable number 4 → KILLED  
 12. Decrementd (--a) double local variable number 2 → KILLED  
 13. Decrementd (--a) double local variable number 4 → KILLED  
 1. negated conditional → KILLED  
 2. removed call to java/lang/Double::isNaN → SURVIVED  
 3. removed conditional - replaced equality check with false → SURVIVED  
 4. removed conditional - replaced equality check with true → KILLED  
 5. Negated double local variable number 0 → SURVIVED  
 6. equal to less than → KILLED  
 7. equal to less or equal → SURVIVED  
 8. equal to greater than → KILLED  
 9. equal to greater or equal → SURVIVED  
 10. equal to not equal → KILLED  
 11. Incremented (a++) double local variable number 0 → KILLED  
 12. Decrementd (a--) double local variable number 0 → KILLED  
 13. Incremented (++a) double local variable number 0 → KILLED  
 14. Decrementd (--a) double local variable number 0 → KILLED  
 1. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED  
 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → KILLED  
 3. Negated double local variable number 2 → SURVIVED  
 4. Incremented (a++) double local variable number 2 → SURVIVED  
 5. Decrementd (a--) double local variable number 2 → SURVIVED  
 6. Incremented (++a) double local variable number 2 → SURVIVED  
 7. Decrementd (--a) double local variable number 2 → SURVIVED  
 1. negated conditional → KILLED  
 2. removed call to java/lang/Double::isNaN → KILLED  
 3. removed conditional - replaced equality check with false → KILLED  
 4. removed conditional - replaced equality check with true → KILLED  
 5. Negated double local variable number 2 → SURVIVED  
 6. equal to less than → KILLED  
 7. equal to less or equal → SURVIVED  
 8. equal to greater than → KILLED  
 9. equal to greater or equal → KILLED  
 10. equal to not equal → KILLED  
 11. Incremented (a++) double local variable number 2 → KILLED  
 12. Decrementd (a--) double local variable number 2 → KILLED  
 13. Incremented (++a) double local variable number 2 → KILLED  
 14. Decrementd (--a) double local variable number 2 → KILLED  
 1. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED  
 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → KILLED  
 3. Negated double local variable number 0 → KILLED  
 4. Incremented (a++) double local variable number 0 → SURVIVED  
 5. Decrementd (a--) double local variable number 0 → SURVIVED  
 6. Incremented (++a) double local variable number 0 → KILLED  
 7. Decrementd (--a) double local variable number 0 → KILLED  
 1. replaced call to java/lang/Math::min with argument → KILLED  
 2. removed call to java/lang/Math::min → KILLED  
 3. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED  
 4. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → KILLED  
 5. Negated double local variable number 0 → KILLED  
 6. Negated double local variable number 2 → KILLED  
 7. Incremented (a++) double local variable number 0 → SURVIVED  
 8. Incremented (a++) double local variable number 2 → SURVIVED  
 9. Decrementd (a--) double local variable number 0 → SURVIVED  
 10. Decrementd (a--) double local variable number 2 → SURVIVED

11. Incremented (++a) double local variable number 0 → KILLED  
 12. Incremented (++a) double local variable number 2 → KILLED  
 13. Decrementd (--a) double local variable number 0 → KILLED  
 14. Decrementd (--a) double local variable number 2 → KILLED  
 1. negated conditional → KILLED  
 2. removed call to java/lang/Double::isNaN → KILLED  
 3. removed conditional - replaced equality check with false → KILLED  
 4. removed conditional - replaced equality check with true → KILLED  
 5. Negated double local variable number 0 → SURVIVED  
 6. equal to less than → KILLED  
 7. equal to less or equal → SURVIVED  
 8. equal to greater than → KILLED  
 9. equal to greater or equal → KILLED  
 10. equal to not equal → KILLED  
 11. Incremented (a++) double local variable number 0 → KILLED  
 12. Decrementd (a--) double local variable number 0 → KILLED  
 13. Incremented (++a) double local variable number 0 → KILLED  
 14. Decrementd (--a) double local variable number 0 → KILLED  
 1. replaced double return with 0.0d for org/jfree/data/Range::max → KILLED  
 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → KILLED  
 3. Negated double local variable number 2 → KILLED  
 4. Incremented (a++) double local variable number 2 → SURVIVED  
 5. Decrementd (a--) double local variable number 2 → SURVIVED  
 6. Incremented (++a) double local variable number 2 → KILLED  
 7. Decrementd (--a) double local variable number 2 → KILLED  
 1. negated conditional → KILLED  
 2. removed call to java/lang/Double::isNaN → SURVIVED  
 3. removed conditional - replaced equality check with false → SURVIVED  
 4. removed conditional - replaced equality check with true → KILLED  
 5. Negated double local variable number 2 → SURVIVED  
 6. equal to less than → KILLED  
 7. equal to less or equal → SURVIVED  
 8. equal to greater than → KILLED  
 9. equal to greater or equal → SURVIVED  
 10. equal to not equal → KILLED  
 11. Incremented (a++) double local variable number 2 → KILLED  
 12. Decrementd (a--) double local variable number 2 → KILLED  
 13. Incremented (++a) double local variable number 2 → KILLED  
 14. Decrementd (--a) double local variable number 2 → KILLED  
 1. replaced double return with 0.0d for org/jfree/data/Range::max → NO\_COVERAGE  
 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → NO\_COVERAGE  
 3. Negated double local variable number 0 → NO\_COVERAGE  
 4. Incremented (a++) double local variable number 0 → NO\_COVERAGE  
 5. Decrementd (a--) double local variable number 0 → NO\_COVERAGE  
 6. Incremented (++a) double local variable number 0 → NO\_COVERAGE  
 7. Decrementd (--a) double local variable number 0 → NO\_COVERAGE  
 1. replaced call to java/lang/Math::max with argument → KILLED  
 2. removed call to java/lang/Math::max → KILLED  
 3. replaced double return with 0.0d for org/jfree/data/Range::max → KILLED  
 4. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → KILLED  
 5. Negated double local variable number 0 → KILLED  
 6. Negated double local variable number 2 → KILLED  
 7. Incremented (a++) double local variable number 0 → SURVIVED  
 8. Incremented (a++) double local variable number 2 → SURVIVED  
 9. Decrementd (a--) double local variable number 0 → SURVIVED  
 10. Decrementd (a--) double local variable number 2 → SURVIVED  
 11. Incremented (++a) double local variable number 0 → KILLED  
 12. Incremented (++a) double local variable number 2 → KILLED  
 13. Decrementd (--a) double local variable number 0 → KILLED  
 14. Decrementd (--a) double local variable number 2 → KILLED  
 1. negated conditional → KILLED  
 2. removed conditional - replaced equality check with false → KILLED

3. removed conditional - replaced equality check with true → KILLED  
 4. not equal to equal → KILLED

1. removed call to org/jfree/data/Range::<init> → KILLED  
 2. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED  
 3. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED  
 4. Negated double local variable number 1 → KILLED  
 5. Negated double local variable number 1 → KILLED

[303](#) 6. Incremented (a++) double local variable number 1 → KILLED  
 7. Incremented (a++) double local variable number 1 → SURVIVED  
 8. Decrementd (a--) double local variable number 1 → KILLED  
 9. Decrementd (a--) double local variable number 1 → SURVIVED  
 10. Incremented (++) double local variable number 1 → KILLED  
 11. Incremented (++) double local variable number 1 → KILLED  
 12. Decrementd (--a) double local variable number 1 → KILLED  
 13. Decrementd (--a) double local variable number 1 → KILLED

1. changed conditional boundary → SURVIVED  
 2. negated conditional → KILLED  
 3. removed call to org/jfree/data/Range::getLowerBound → KILLED  
 4. removed conditional - replaced comparison check with false → KILLED  
 5. removed conditional - replaced comparison check with true → KILLED  
 6. Negated double local variable number 1 → KILLED

[305](#) 7. greater or equal to less than → KILLED  
 8. greater or equal to less or equal → KILLED  
 9. greater or equal to greater than → SURVIVED  
 10. greater or equal to equal → KILLED  
 11. greater or equal to not equal → KILLED  
 12. Incremented (a++) double local variable number 1 → KILLED  
 13. Decrementd (a--) double local variable number 1 → KILLED  
 14. Incremented (++) double local variable number 1 → KILLED  
 15. Decrementd (--a) double local variable number 1 → KILLED

1. removed call to org/jfree/data/Range::<init> → KILLED  
 2. removed call to org/jfree/data/Range::getUpperBound → KILLED  
 3. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED  
 4. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED

[306](#) 5. Negated double local variable number 1 → KILLED  
 6. Incremented (a++) double local variable number 1 → SURVIVED  
 7. Decrementd (a--) double local variable number 1 → SURVIVED  
 8. Incremented (++) double local variable number 1 → KILLED  
 9. Decrementd (--a) double local variable number 1 → KILLED

1. changed conditional boundary → SURVIVED  
 2. negated conditional → KILLED  
 3. removed call to org/jfree/data/Range::getUpperBound → KILLED  
 4. removed conditional - replaced comparison check with false → KILLED  
 5. removed conditional - replaced comparison check with true → KILLED  
 6. Negated double local variable number 1 → KILLED

[308](#) 7. Less or equal to less than → SURVIVED  
 8. Less or equal to greater than → KILLED  
 9. Less or equal to greater or equal → KILLED  
 10. Less or equal to equal → KILLED  
 11. Less or equal to not equal → KILLED  
 12. Incremented (a++) double local variable number 1 → KILLED  
 13. Decrementd (a--) double local variable number 1 → KILLED  
 14. Incremented (++) double local variable number 1 → KILLED  
 15. Decrementd (--a) double local variable number 1 → KILLED

[309](#) 1. removed call to org/jfree/data/Range::<init> → KILLED  
 2. removed call to org/jfree/data/Range::getLowerBound → KILLED  
 3. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED  
 4. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED  
 5. Negated double local variable number 1 → KILLED

6. Incremented (a++) double local variable number 1 → SURVIVED  
7. Decrementd (a--) double local variable number 1 → SURVIVED  
8. Incremented (++a) double local variable number 1 → KILLED  
9. Decrementd (--a) double local variable number 1 → KILLED  
1. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED  
312 2. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED  
329 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED  
330 1. removed call to org/jfree/data/Range::getLength → KILLED  
1. Replaced double multiplication with division → KILLED  
2. Replaced double subtraction with addition → KILLED  
3. removed call to org/jfree/data/Range::getLowerBound → KILLED  
4. Negated double local variable number 5 → KILLED  
5. Negated double local variable number 1 → KILLED  
6. Replaced double operation with first member → KILLED  
7. Replaced double operation with first member → KILLED  
8. Replaced double operation by second member → KILLED  
9. Replaced double operation by second member → KILLED  
10. Replaced double multiplication with division → KILLED  
11. Replaced double subtraction with addition → KILLED  
12. Replaced double multiplication with modulus → KILLED  
331 13. Replaced double subtraction with multiplication → KILLED  
14. Replaced double multiplication with addition → KILLED  
15. Replaced double subtraction with division → KILLED  
16. Replaced double multiplication with subtraction → KILLED  
17. Replaced double subtraction with modulus → KILLED  
18. Incremented (a++) double local variable number 5 → KILLED  
19. Incremented (a++) double local variable number 1 → SURVIVED  
20. Decrementd (a--) double local variable number 5 → KILLED  
21. Decrementd (a--) double local variable number 1 → SURVIVED  
22. Incremented (++a) double local variable number 5 → KILLED  
23. Incremented (++a) double local variable number 1 → KILLED  
24. Decrementd (--a) double local variable number 5 → KILLED  
25. Decrementd (--a) double local variable number 1 → KILLED  
1. Replaced double multiplication with division → KILLED  
2. Replaced double addition with subtraction → KILLED  
3. removed call to org/jfree/data/Range::getUpperBound → KILLED  
4. Negated double local variable number 5 → KILLED  
5. Negated double local variable number 3 → KILLED  
6. Replaced double operation with first member → KILLED  
7. Replaced double operation with first member → KILLED  
8. Replaced double operation by second member → KILLED  
9. Replaced double operation by second member → KILLED  
10. Replaced double multiplication with division → KILLED  
11. Replaced double addition with subtraction → KILLED  
12. Replaced double multiplication with modulus → KILLED  
332 13. Replaced double addition with multiplication → KILLED  
14. Replaced double multiplication with addition → KILLED  
15. Replaced double addition with division → KILLED  
16. Replaced double multiplication with subtraction → KILLED  
17. Replaced double addition with modulus → KILLED  
18. Incremented (a++) double local variable number 5 → SURVIVED  
19. Incremented (a++) double local variable number 3 → SURVIVED  
20. Decrementd (a--) double local variable number 5 → SURVIVED  
21. Decrementd (a--) double local variable number 3 → SURVIVED  
22. Incremented (++a) double local variable number 5 → KILLED  
23. Incremented (++a) double local variable number 3 → KILLED  
24. Decrementd (--a) double local variable number 5 → KILLED  
25. Decrementd (--a) double local variable number 3 → KILLED  
333 1. changed conditional boundary → SURVIVED  
2. negated conditional → KILLED  
3. removed conditional - replaced comparison check with false → KILLED  
4. removed conditional - replaced comparison check with true → KILLED  
5. Negated double local variable number 7 → KILLED



6. Negated double local variable number 9 → KILLED  
 7. Less or equal to less than → SURVIVED  
 8. Less or equal to greater than → KILLED  
 9. Less or equal to greater or equal → KILLED  
 10. Less or equal to equal → KILLED  
 11. Less or equal to not equal → KILLED  
 12. Incremented (a++) double local variable number 7 → KILLED  
 13. Incremented (a++) double local variable number 9 → KILLED  
 14. Decrementd (a--) double local variable number 7 → KILLED  
 15. Decrementd (a--) double local variable number 9 → KILLED  
 16. Incremented (++a) double local variable number 7 → KILLED  
 17. Incremented (++a) double local variable number 9 → KILLED  
 18. Decrementd (--a) double local variable number 7 → KILLED  
 19. Decrementd (--a) double local variable number 9 → KILLED  
 1. Substituted 2.0 with 1.0 → KILLED  
 2. Substituted 2.0 with 1.0 → KILLED  
 3. Replaced double division with multiplication → KILLED  
 4. Replaced double division with multiplication → KILLED  
 5. Replaced double addition with subtraction → KILLED  
 6. Negated double local variable number 7 → KILLED  
 7. Negated double local variable number 9 → KILLED  
 8. Replaced double operation with first member → KILLED  
 9. Replaced double operation with first member → KILLED  
 10. Replaced double operation with first member → KILLED  
 11. Replaced double operation by second member → KILLED  
 12. Replaced double operation by second member → KILLED  
 13. Replaced double operation by second member → KILLED  
 14. Replaced double division with multiplication → KILLED  
 15. Replaced double division with multiplication → KILLED  
 16. Replaced double addition with subtraction → KILLED  
 17. Replaced double division with modulus → KILLED  
 18. Replaced double division with modulus → KILLED  
 19. Replaced double addition with multiplication → KILLED  
 20. Replaced double division with addition → KILLED  
 21. Replaced double division with addition → KILLED  
 22. Replaced double addition with division → KILLED  
 23. Replaced double division with subtraction → KILLED  
 24. Replaced double division with subtraction → KILLED  
 25. Replaced double addition with modulus → KILLED  
 26. Substituted 2.0 with 1.0 → KILLED  
 27. Substituted 2.0 with 1.0 → KILLED  
 28. Substituted 2.0 with 0.0 → KILLED  
 29. Substituted 2.0 with 0.0 → KILLED  
 30. Substituted 2.0 with -1.0 → KILLED  
 31. Substituted 2.0 with -1.0 → KILLED  
 32. Substituted 2.0 with -2.0 → KILLED  
 33. Substituted 2.0 with -2.0 → KILLED  
 34. Substituted 2.0 with 3.0 → KILLED  
 35. Substituted 2.0 with 3.0 → KILLED  
 36. Substituted 2.0 with 1.0 → KILLED  
 37. Substituted 2.0 with 1.0 → KILLED  
 38. Incremented (a++) double local variable number 7 → SURVIVED  
 39. Incremented (a++) double local variable number 9 → SURVIVED  
 40. Decrementd (a--) double local variable number 7 → SURVIVED  
 41. Decrementd (a--) double local variable number 9 → SURVIVED  
 42. Incremented (++a) double local variable number 7 → KILLED  
 43. Incremented (++a) double local variable number 9 → KILLED  
 44. Decrementd (--a) double local variable number 7 → KILLED  
 45. Decrementd (--a) double local variable number 9 → KILLED  
 1. Negated double local variable number 7 → KILLED  
 2. Incremented (a++) double local variable number 7 → KILLED  
 3. Decrementd (a--) double local variable number 7 → KILLED  
 4. Incremented (++a) double local variable number 7 → KILLED  
 5. Decrementd (--a) double local variable number 7 → KILLED  
 1. removed call to org/jfree/data/Range::<init> → KILLED  
 2. replaced return value with null for org/jfree/data/Range::expand → KILLED

3. mutated return of Object value for org/jfree/data/Range::expand to ( if (x != null) null else throw new RuntimeException ) → KILLED

4. Negated double local variable number 7 → KILLED

5. Negated double local variable number 9 → KILLED

6. Incremented (a++) double local variable number 7 → SURVIVED

7. Incremented (a++) double local variable number 9 → SURVIVED

8. Decrementd (a--) double local variable number 7 → SURVIVED

9. Decrementd (a--) double local variable number 9 → SURVIVED

10. Incremented (++) double local variable number 7 → KILLED

11. Incremented (++) double local variable number 9 → KILLED

12. Decrementd (--a) double local variable number 7 → KILLED

13. Decrementd (--a) double local variable number 9 → KILLED

1. replaced call to org/jfree/data/Range::shift with argument → KILLED

2. Substituted 0 with 1 → KILLED

3. removed call to org/jfree/data/Range::shift → KILLED

4. replaced return value with null for org/jfree/data/Range::shift → KILLED

5. mutated return of Object value for org/jfree/data/Range::shift to ( if (x != null) null else throw new RuntimeException ) → KILLED

349 6. Negated double local variable number 1 → KILLED

7. Substituted 0 with 1 → KILLED

8. Substituted 0 with -1 → KILLED

9. Substituted 0 with 1 → KILLED

10. Substituted 0 with -1 → KILLED

11. Incremented (a++) double local variable number 1 → SURVIVED

12. Decrementd (a--) double local variable number 1 → SURVIVED

13. Incremented (++) double local variable number 1 → KILLED

14. Decrementd (--a) double local variable number 1 → KILLED

365 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED

1. negated conditional → KILLED

2. removed conditional - replaced equality check with false → KILLED

3. removed conditional - replaced equality check with true → KILLED

4. Negated integer local variable number 3 → SURVIVED

5. equal to less than → KILLED

6. equal to less or equal → SURVIVED

366 7. equal to greater than → KILLED

8. equal to greater or equal → KILLED

9. equal to not equal → KILLED

10. Incremented (a++) integer local variable number 3 → SURVIVED

11. Decrementd (a--) integer local variable number 3 → SURVIVED

12. Incremented (++) integer local variable number 3 → KILLED

13. Decrementd (--a) integer local variable number 3 → KILLED

1. removed call to org/jfree/data/Range::<init> → KILLED

2. Replaced double addition with subtraction → KILLED

3. removed call to org/jfree/data/Range::getLowerBound → KILLED

4. replaced return value with null for org/jfree/data/Range::shift → KILLED

5. mutated return of Object value for org/jfree/data/Range::shift to ( if (x != null) null else throw new RuntimeException ) → KILLED

6. Negated double local variable number 1 → KILLED

7. Replaced double operation with first member → KILLED

367 8. Replaced double operation by second member → KILLED

9. Replaced double addition with subtraction → KILLED

10. Replaced double addition with multiplication → KILLED

11. Replaced double addition with division → KILLED

12. Replaced double addition with modulus → KILLED

13. Incremented (a++) double local variable number 1 → KILLED

14. Decrementd (a--) double local variable number 1 → KILLED

15. Incremented (++) double local variable number 1 → KILLED

16. Decrementd (--a) double local variable number 1 → KILLED

368 1. Replaced double addition with subtraction → KILLED

2. removed call to org/jfree/data/Range::getUpperBound → KILLED

3. Negated double local variable number 1 → KILLED

4. Replaced double operation with first member → KILLED

5. Replaced double operation by second member → KILLED

6. Replaced double addition with subtraction → KILLED

7. Replaced double addition with multiplication → KILLED

8. Replaced double addition with division → KILLED  
 9. Replaced double addition with modulus → KILLED  
 10. Incremented (a++) double local variable number 1 → SURVIVED  
 11. Decrementd (a--) double local variable number 1 → SURVIVED  
 12. Incremented (++a) double local variable number 1 → KILLED  
 13. Decrementd (--a) double local variable number 1 → KILLED  
 1. replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument → KILLED  
 2. removed call to org/jfree/data/Range::<init> → KILLED  
 371 3. removed call to org/jfree/data/Range::getLowerBound → KILLED  
 4. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
 5. replaced return value with null for org/jfree/data/Range::shift → KILLED  
 6. mutated return of Object value for org/jfree/data/Range::shift to ( if (x != null) null else throw new RuntimeException ) → KILLED  
 1. replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument → KILLED  
 2. removed call to org/jfree/data/Range::getUpperBound → KILLED  
 372 3. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
 4. Negated double local variable number 1 → KILLED  
 5. Incremented (a++) double local variable number 1 → KILLED  
 6. Decrementd (a--) double local variable number 1 → KILLED  
 7. Incremented (++a) double local variable number 1 → KILLED  
 8. Decrementd (--a) double local variable number 1 → KILLED  
 1. Negated double local variable number 1 → KILLED  
 2. Incremented (a++) double local variable number 1 → SURVIVED  
 373 3. Decrementd (a--) double local variable number 1 → SURVIVED  
 4. Incremented (++a) double local variable number 1 → KILLED  
 5. Decrementd (--a) double local variable number 1 → KILLED  
 1. changed conditional boundary → SURVIVED  
 2. Substituted 0.0 with 1.0 → SURVIVED  
 3. negated conditional → KILLED  
 4. removed conditional - replaced comparison check with false → KILLED  
 5. removed conditional - replaced comparison check with true → KILLED  
 6. Negated double local variable number 0 → KILLED  
 7. Substituted 0.0 with 1.0 → SURVIVED  
 8. Substituted 0.0 with -1.0 → SURVIVED  
 9. Substituted 0.0 with 1.0 → SURVIVED  
 387 10. Substituted 0.0 with -1.0 → SURVIVED  
 11. Less or equal to less than → SURVIVED  
 12. Less or equal to greater than → KILLED  
 13. Less or equal to greater or equal → KILLED  
 14. Less or equal to equal → KILLED  
 15. Less or equal to not equal → KILLED  
 16. Incremented (a++) double local variable number 0 → KILLED  
 17. Decrementd (a--) double local variable number 0 → KILLED  
 18. Incremented (++a) double local variable number 0 → KILLED  
 19. Decrementd (--a) double local variable number 0 → KILLED  
 388 1. replaced call to java/lang/Math::max with argument → KILLED  
 2. Substituted 0.0 with 1.0 → KILLED  
 3. Replaced double addition with subtraction → KILLED  
 4. removed call to java/lang/Math::max → KILLED  
 5. replaced double return with 0.0d for  
 org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
 6. replaced return of double value with -(x + 1) for  
 org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
 7. Negated double local variable number 0 → KILLED  
 8. Negated double local variable number 2 → KILLED  
 9. Replaced double operation with first member → KILLED  
 10. Replaced double operation by second member → KILLED  
 11. Replaced double addition with subtraction → KILLED  
 12. Replaced double addition with multiplication → KILLED  
 13. Replaced double addition with division → KILLED  
 14. Replaced double addition with modulus → KILLED  
 15. Substituted 0.0 with 1.0 → KILLED  
 16. Substituted 0.0 with -1.0 → KILLED



17. Substituted 0.0 with 1.0 → KILLED  
 18. Substituted 0.0 with -1.0 → KILLED  
 19. Incremented (a++) double local variable number 0 → SURVIVED  
 20. Incremented (a++) double local variable number 2 → SURVIVED  
 21. Decrementd (a--) double local variable number 0 → SURVIVED  
 22. Decrementd (a--) double local variable number 2 → SURVIVED  
 23. Incremented (++a) double local variable number 0 → KILLED  
 24. Incremented (++a) double local variable number 2 → KILLED  
 25. Decrementd (--a) double local variable number 0 → KILLED  
 26. Decrementd (--a) double local variable number 2 → KILLED  
 1. changed conditional boundary → SURVIVED  
 2. Substituted 0.0 with 1.0 → SURVIVED  
 3. negated conditional → KILLED  
 4. removed conditional - replaced comparison check with false → KILLED  
 5. removed conditional - replaced comparison check with true → SURVIVED  
 6. Negated double local variable number 0 → KILLED  
 7. Substituted 0.0 with 1.0 → SURVIVED  
 8. Substituted 0.0 with -1.0 → SURVIVED  
 9. Substituted 0.0 with 1.0 → SURVIVED  
 390 10. Substituted 0.0 with -1.0 → SURVIVED  
 11. greater or equal to less than → KILLED  
 12. greater or equal to less or equal → KILLED  
 13. greater or equal to greater than → SURVIVED  
 14. greater or equal to equal → SURVIVED  
 15. greater or equal to not equal → KILLED  
 16. Incremented (a++) double local variable number 0 → KILLED  
 17. Decrementd (a--) double local variable number 0 → KILLED  
 18. Incremented (++a) double local variable number 0 → KILLED  
 19. Decrementd (--a) double local variable number 0 → KILLED  
 1. replaced call to java/lang/Math::min with argument → KILLED  
 2. Substituted 0.0 with 1.0 → KILLED  
 3. Replaced double addition with subtraction → KILLED  
 4. removed call to java/lang/Math::min → KILLED  
 5. replaced double return with 0.0d for  
 org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
 6. replaced return of double value with -(x + 1) for  
 org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
 7. Negated double local variable number 0 → KILLED  
 8. Negated double local variable number 2 → KILLED  
 9. Replaced double operation with first member → KILLED  
 10. Replaced double operation by second member → KILLED  
 11. Replaced double addition with subtraction → KILLED  
 391 12. Replaced double addition with multiplication → KILLED  
 13. Replaced double addition with division → KILLED  
 14. Replaced double addition with modulus → KILLED  
 15. Substituted 0.0 with 1.0 → KILLED  
 16. Substituted 0.0 with -1.0 → KILLED  
 17. Substituted 0.0 with 1.0 → KILLED  
 18. Substituted 0.0 with -1.0 → KILLED  
 19. Incremented (a++) double local variable number 0 → SURVIVED  
 20. Incremented (a++) double local variable number 2 → SURVIVED  
 21. Decrementd (a--) double local variable number 0 → SURVIVED  
 22. Decrementd (a--) double local variable number 2 → SURVIVED  
 23. Incremented (++a) double local variable number 0 → KILLED  
 24. Incremented (++a) double local variable number 2 → KILLED  
 25. Decrementd (--a) double local variable number 0 → KILLED  
 26. Decrementd (--a) double local variable number 2 → KILLED  
 394 1. Replaced double addition with subtraction → NO\_COVERAGE  
 2. replaced double return with 0.0d for  
 org/jfree/data/Range::shiftWithNoZeroCrossing → NO\_COVERAGE  
 3. replaced return of double value with -(x + 1) for  
 org/jfree/data/Range::shiftWithNoZeroCrossing → NO\_COVERAGE  
 4. Negated double local variable number 0 → NO\_COVERAGE  
 5. Negated double local variable number 2 → NO\_COVERAGE  
 6. Replaced double operation with first member → NO\_COVERAGE  
 7. Replaced double operation by second member → NO\_COVERAGE

```

8. Replaced double addition with subtraction → NO_COVERAGE
9. Replaced double addition with multiplication → NO_COVERAGE
10. Replaced double addition with division → NO_COVERAGE
11. Replaced double addition with modulus → NO_COVERAGE
12. Incremented (a++) double local variable number 0 → NO_COVERAGE
13. Incremented (a++) double local variable number 2 → NO_COVERAGE
14. Decrementd (a--) double local variable number 0 → NO_COVERAGE
15. Decrementd (a--) double local variable number 2 → NO_COVERAGE
16. Incremented (++a) double local variable number 0 → NO_COVERAGE
17. Incremented (++a) double local variable number 2 → NO_COVERAGE
18. Decrementd (--a) double local variable number 0 → NO_COVERAGE
19. Decrementd (--a) double local variable number 2 → NO_COVERAGE
409 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
    1. changed conditional boundary → KILLED
    2. Substituted 0.0 with 1.0 → KILLED
    3. negated conditional → KILLED
    4. removed conditional - replaced comparison check with false → KILLED
    5. removed conditional - replaced comparison check with true → KILLED
    6. Negated double local variable number 1 → KILLED
    7. Substituted 0.0 with 1.0 → KILLED
    8. Substituted 0.0 with -1.0 → KILLED
    9. Substituted 0.0 with 1.0 → KILLED
410 10. Substituted 0.0 with -1.0 → KILLED
    11. greater or equal to less than → KILLED
    12. greater or equal to less or equal → KILLED
    13. greater or equal to greater than → KILLED
    14. greater or equal to equal → KILLED
    15. greater or equal to not equal → KILLED
    16. Incremented (a++) double local variable number 1 → KILLED
    17. Decrementd (a--) double local variable number 1 → KILLED
    18. Incremented (++a) double local variable number 1 → KILLED
    19. Decrementd (--a) double local variable number 1 → KILLED
411 1. removed call to java/lang/IllegalArgumentException::<init> → KILLED
    1. removed call to org/jfree/data/Range::<init> → KILLED
    2. Replaced double multiplication with division → KILLED
    3. removed call to org/jfree/data/Range::getLowerBound → KILLED
    4. replaced return value with null for org/jfree/data/Range::scale → KILLED
    5. mutated return of Object value for org/jfree/data/Range::scale to ( if (x !=
    null) null else throw new RuntimeException ) → KILLED
    6. Negated double local variable number 1 → KILLED
    7. Replaced double operation with first member → KILLED
413 8. Replaced double operation by second member → KILLED
    9. Replaced double multiplication with division → KILLED
    10. Replaced double multiplication with modulus → KILLED
    11. Replaced double multiplication with addition → KILLED
    12. Replaced double multiplication with subtraction → KILLED
    13. Incremented (a++) double local variable number 1 → KILLED
    14. Decrementd (a--) double local variable number 1 → KILLED
    15. Incremented (++a) double local variable number 1 → KILLED
    16. Decrementd (--a) double local variable number 1 → KILLED
    1. Replaced double multiplication with division → KILLED
    2. removed call to org/jfree/data/Range::getUpperBound → KILLED
    3. Negated double local variable number 1 → KILLED
    4. Replaced double operation with first member → KILLED
    5. Replaced double operation by second member → KILLED
    6. Replaced double multiplication with division → KILLED
414 7. Replaced double multiplication with modulus → KILLED
    8. Replaced double multiplication with addition → KILLED
    9. Replaced double multiplication with subtraction → KILLED
    10. Incremented (a++) double local variable number 1 → SURVIVED
    11. Decrementd (a--) double local variable number 1 → SURVIVED
    12. Incremented (++a) double local variable number 1 → KILLED
    13. Decrementd (--a) double local variable number 1 → KILLED
426 1. negated conditional → KILLED
    2. removed conditional - replaced equality check with false → KILLED

```

3. removed conditional - replaced equality check with true → KILLED  
 4. not equal to less than → KILLED  
 5. not equal to less or equal → KILLED  
 6. not equal to greater than → SURVIVED  
 7. not equal to greater or equal → KILLED  
 8. not equal to equal → KILLED  
 1. replaced boolean return with true for org/jfree/data/Range::equals → KILLED  
 2. Substituted 0 with 1 → KILLED  
 3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 427 4. Substituted 0 with 1 → KILLED  
 5. Substituted 0 with -1 → KILLED  
 6. Substituted 0 with 1 → KILLED  
 7. Substituted 0 with -1 → KILLED  
 1. negated conditional → KILLED  
 2. removed conditional - replaced equality check with false → KILLED  
 3. removed conditional - replaced equality check with true → KILLED  
 4. Negated double field lower → KILLED  
 5. Negated double field lower → KILLED  
 6. equal to less than → KILLED  
 7. equal to less or equal → SURVIVED  
 8. equal to greater than → KILLED  
 9. equal to greater or equal → KILLED  
 430 10. equal to not equal → KILLED  
 11. Incremented (a++) double field lower → KILLED  
 12. Incremented (a++) double field lower → KILLED  
 13. Decrementd (a--) double field lower → KILLED  
 14. Decrementd (a--) double field lower → KILLED  
 15. Incremented (++a) double field lower → KILLED  
 16. Incremented (++a) double field lower → KILLED  
 17. Decrementd (--a) double fieldlower → KILLED  
 18. Decrementd (--a) double fieldlower → KILLED  
 1. replaced boolean return with true for org/jfree/data/Range::equals → KILLED  
 2. Substituted 0 with 1 → KILLED  
 3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 431 4. Substituted 0 with 1 → KILLED  
 5. Substituted 0 with -1 → KILLED  
 6. Substituted 0 with 1 → KILLED  
 7. Substituted 0 with -1 → KILLED  
 1. negated conditional → KILLED  
 2. removed conditional - replaced equality check with false → KILLED  
 3. removed conditional - replaced equality check with true → KILLED  
 4. Negated double field upper → KILLED  
 5. Negated double field upper → KILLED  
 6. equal to less than → KILLED  
 7. equal to less or equal → SURVIVED  
 8. equal to greater than → KILLED  
 9. equal to greater or equal → KILLED  
 433 10. equal to not equal → KILLED  
 11. Incremented (a++) double field upper → KILLED  
 12. Incremented (a++) double field upper → KILLED  
 13. Decrementd (a--) double field upper → KILLED  
 14. Decrementd (a--) double field upper → KILLED  
 15. Incremented (++a) double field upper → KILLED  
 16. Incremented (++a) double field upper → KILLED  
 17. Decrementd (--a) double fieldupper → KILLED  
 18. Decrementd (--a) double fieldupper → KILLED  
 1. replaced boolean return with true for org/jfree/data/Range::equals → KILLED  
 2. Substituted 0 with 1 → KILLED  
 3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 434 4. Substituted 0 with 1 → KILLED  
 5. Substituted 0 with -1 → KILLED  
 6. Substituted 0 with 1 → KILLED  
 7. Substituted 0 with -1 → KILLED  
 436 1. replaced boolean return with false for org/jfree/data/Range::equals → KILLED  
 2. Substituted 1 with 0 → KILLED

3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 4. Substituted 1 with 0 → KILLED  
 5. Substituted 1 with -1 → SURVIVED  
 6. Substituted 1 with -1 → SURVIVED  
 7. Substituted 1 with 2 → KILLED  
 8. Substituted 1 with 0 → KILLED  
 1. replaced boolean return with false for org/jfree/data/Range::isNaNRange → KILLED  
 2. replaced boolean return with true for org/jfree/data/Range::isNaNRange → KILLED  
 3. Substituted 1 with 0 → KILLED  
 4. Substituted 0 with 1 → KILLED  
 5. negated conditional → KILLED  
 6. negated conditional → KILLED  
 7. removed call to java/lang/Double::isNaN → KILLED  
 8. removed call to java/lang/Double::isNaN → KILLED  
 9. removed conditional - replaced equality check with false → KILLED  
 10. removed conditional - replaced equality check with false → KILLED  
 11. removed conditional - replaced equality check with true → SURVIVED  
 12. removed conditional - replaced equality check with true → SURVIVED  
 13. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 15. Negated double field lower → SURVIVED  
 16. Negated double field upper → SURVIVED  
 17. Substituted 0 with 1 → KILLED  
 18. Substituted 1 with 0 → KILLED  
 19. Substituted 1 with -1 → SURVIVED  
 20. Substituted 0 with -1 → KILLED  
 448 21. Substituted 1 with -1 → SURVIVED  
 22. Substituted 1 with 2 → KILLED  
 23. Substituted 0 with 1 → KILLED  
 24. Substituted 1 with 0 → KILLED  
 25. Substituted 0 with -1 → KILLED  
 26. equal to less than → SURVIVED  
 27. equal to less than → SURVIVED  
 28. equal to less or equal → SURVIVED  
 29. equal to less or equal → SURVIVED  
 30. equal to greater than → KILLED  
 31. equal to greater than → KILLED  
 32. equal to greater or equal → KILLED  
 33. equal to greater or equal → KILLED  
 34. equal to not equal → KILLED  
 35. equal to not equal → KILLED  
 36. Incremented (a++) double field lower → SURVIVED  
 37. Incremented (a++) double field upper → SURVIVED  
 38. Decrementd (a--) double field lower → SURVIVED  
 39. Decrementd (a--) double field upper → SURVIVED  
 40. Incremented (++a) double field lower → SURVIVED  
 41. Incremented (++a) double field upper → SURVIVED  
 42. Decrementd (--a) double fieldlower → SURVIVED  
 43. Decrementd (--a) double fieldupper → SURVIVED  
 1. removed call to java/lang/Double::doubleToLongBits → KILLED  
 2. Negated double field lower → KILLED  
 460 3. Incremented (a++) double field lower → SURVIVED  
 4. Decrementd (a--) double field lower → SURVIVED  
 5. Incremented (++a) double field lower → KILLED  
 6. Decrementd (--a) double fieldlower → KILLED  
 461 1. Substituted 32 with 33 → KILLED  
 2. Replaced Unsigned Shift Right with Shift Left → KILLED  
 3. Replaced XOR with AND → KILLED  
 4. Negated long local variable number 1 → SURVIVED  
 5. Negated long local variable number 1 → KILLED  
 6. Substituted 32 with 1 → KILLED  
 7. Substituted 32 with 0 → KILLED  
 8. Substituted 32 with -1 → KILLED  
 9. Substituted 32 with -32 → SURVIVED

10. Substituted 32 with 33 → KILLED  
 11. Substituted 32 with 31 → KILLED  
 12. Incremented (a++) long local variable number 1 → SURVIVED  
 13. Incremented (a++) long local variable number 1 → SURVIVED  
 14. Decrementd (a--) long local variable number 2 → KILLED  
 15. Decrementd (a--) long local variable number 2 → SURVIVED  
 16. Incremented (++a) long local variable number 1 → KILLED  
 17. Incremented (++a) long local variable number 1 → SURVIVED  
 18. Decrementd (--a) long local variable number 2 → KILLED  
 19. Decrementd (--a) long local variable number 2 → KILLED  
 1. removed call to java/lang/Double::doubleToLongBits → KILLED  
 2. Negated double field upper → KILLED  
 462 3. Incremented (a++) double field upper → SURVIVED  
 4. Decrementd (a--) double field upper → SURVIVED  
 5. Incremented (++a) double field upper → KILLED  
 6. Decrementd (--a) double fieldupper → KILLED  
 1. Substituted 29 with 30 → KILLED  
 2. Substituted 32 with 33 → KILLED  
 3. Replaced integer multiplication with division → KILLED  
 4. Replaced Unsigned Shift Right with Shift Left → KILLED  
 5. Replaced XOR with AND → KILLED  
 6. Replaced integer addition with subtraction → KILLED  
 7. Negated integer local variable number 3 → KILLED  
 8. Negated long local variable number 1 → SURVIVED  
 9. Negated long local variable number 1 → KILLED  
 10. Replaced integer operation with first member → KILLED  
 11. Replaced integer operation with first member → KILLED  
 12. Replaced integer operation by second member → SURVIVED  
 13. Replaced integer operation by second member → KILLED  
 14. Replaced integer multiplication with division → KILLED  
 15. Replaced integer addition with subtraction → KILLED  
 16. Replaced integer multiplication with modulus → KILLED  
 17. Replaced integer addition with multiplication → KILLED  
 18. Replaced integer multiplication with addition → KILLED  
 19. Replaced integer addition with division → KILLED  
 20. Replaced integer multiplication with subtraction → KILLED  
 21. Replaced integer addition with modulus → KILLED  
 463 22. Substituted 29 with 1 → SURVIVED  
 23. Substituted 32 with 1 → KILLED  
 24. Substituted 29 with 0 → KILLED  
 25. Substituted 32 with 0 → KILLED  
 26. Substituted 29 with -1 → KILLED  
 27. Substituted 32 with -1 → KILLED  
 28. Substituted 29 with -29 → KILLED  
 29. Substituted 32 with -32 → SURVIVED  
 30. Substituted 29 with 30 → KILLED  
 31. Substituted 32 with 33 → KILLED  
 32. Substituted 29 with 28 → KILLED  
 33. Substituted 32 with 31 → KILLED  
 34. Incremented (a++) integer local variable number 3 → SURVIVED  
 35. Incremented (a++) long local variable number 1 → SURVIVED  
 36. Incremented (a++) long local variable number 1 → SURVIVED  
 37. Decrementd (a--) integer local variable number 1 → SURVIVED  
 38. Decrementd (a--) long local variable number 2 → KILLED  
 39. Decrementd (a--) long local variable number 2 → SURVIVED  
 40. Incremented (++a) integer local variable number 3 → KILLED  
 41. Incremented (++a) long local variable number 1 → KILLED  
 42. Incremented (++a) long local variable number 1 → SURVIVED  
 43. Decrementd (--a) integer local variable number 1 → KILLED  
 44. Decrementd (--a) long local variable number 2 → KILLED  
 45. Decrementd (--a) long local variable number 2 → KILLED  
 464 1. replaced int return with 0 for org/jfree/data/Range::hashCode → KILLED  
 2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
 3. Negated integer local variable number 3 → KILLED  
 4. Incremented (a++) integer local variable number 3 → SURVIVED  
 5. Decrementd (a--) integer local variable number 1 → SURVIVED



475

```

6. Incremented (++a) integer local variable number 3 → KILLED
7. Decrementd (--a) integer local variable number 1 → KILLED
1. removed call to java/lang/StringBuilder::<init> → KILLED
2. replaced return value with "" for org/jfree/data/Range::toString → KILLED
3. removed call to java/lang/StringBuilder::append → KILLED
4. removed call to java/lang/StringBuilder::append → KILLED
5. removed call to java/lang/StringBuilder::append → KILLED
6. removed call to java/lang/StringBuilder::append → KILLED
7. removed call to java/lang/StringBuilder::toString → KILLED
8. mutated return of Object value for org/jfree/data/Range::toString to ( if
(x != null) null else throw new RuntimeException ) → KILLED
9. replaced call to java/lang/StringBuilder::append with receiver → KILLED
10. replaced call to java/lang/StringBuilder::append with receiver → KILLED
11. replaced call to java/lang/StringBuilder::append with receiver → KILLED
12. replaced call to java/lang/StringBuilder::append with receiver → KILLED
13. Negated double field lower → KILLED
14. Negated double field upper → KILLED
15. Incremented (a++) double field lower → SURVIVED
16. Incremented (a++) double field upper → SURVIVED
17. Decrementd (a--) double field lower → SURVIVED
18. Decrementd (a--) double field upper → SURVIVED
19. Incremented (++a) double field lower → KILLED
20. Incremented (++a) double field upper → KILLED
21. Decrementd (--a) double fieldlower → KILLED
22. Decrementd (--a) double fieldupper → KILLED

```

## Active mutators

- ABS\_MUTATOR
- AOD\_1\_MUTATOR
- AOD\_2\_MUTATOR
- AOR\_1\_MUTATOR
- AOR\_2\_MUTATOR
- AOR\_3\_MUTATOR
- AOR\_4\_MUTATOR
- ARGUMENT\_PROPAGATION\_MUTATOR
- BOOLEAN\_FALSE\_RETURN
- BOOLEAN\_TRUE\_RETURN
- CONDITIONALS\_BOUNDARY\_MUTATOR
- CONSTRUCTOR\_CALL\_MUTATOR
- CRCR\_1\_MUTATOR
- CRCR\_2\_MUTATOR
- CRCR\_3\_MUTATOR
- CRCR\_4\_MUTATOR
- CRCR\_5\_MUTATOR
- CRCR\_6\_MUTATOR
- EMPTY\_RETURN\_VALUES
- EXPERIMENTAL\_BIGINTEGER\_MUTATOR
- EXPERIMENTAL\_MEMBER\_VARIABLE\_MUTATOR
- EXPERIMENTAL\_REMOVE\_SWITCH\_MUTATOR\_[0-99]
- EXPERIMENTAL\_SWITCH\_MUTATOR
- INCREMENTS\_MUTATOR
- INLINE\_CONSTANT\_MUTATOR
- INVERT\_NEGS\_MUTATOR
- MATH\_MUTATOR
- NAKED\_RECEIVER
- NEGATE\_CONDITIONALS\_MUTATOR
- NON\_VOID\_METHOD\_CALL\_MUTATOR
- NULL\_RETURN\_VALUES
- OBBN\_1\_MUTATOR
- OBBN\_2\_MUTATOR
- OBBN\_3\_MUTATOR
- PRIMITIVE\_RETURN\_VALS\_MUTATOR
- REMOVE\_CONDITIONALS\_EQUAL\_ELSE\_MUTATOR
- REMOVE\_CONDITIONALS\_EQUAL\_IF\_MUTATOR

- REMOVE\_CONDITIONALS\_ORDER\_ELSE\_MUTATOR
- REMOVE\_CONDITIONALS\_ORDER\_IF\_MUTATOR
- REMOVE\_INCREMENTS\_MUTATOR
- RETURN\_VALS\_MUTATOR
- ROR\_1\_MUTATOR
- ROR\_2\_MUTATOR
- ROR\_3\_MUTATOR
- ROR\_4\_MUTATOR
- ROR\_5\_MUTATOR
- UOI\_1\_MUTATOR
- UOI\_2\_MUTATOR
- UOI\_3\_MUTATOR
- UOI\_4\_MUTATOR
- VOID\_METHOD\_CALL\_MUTATOR

## Tests examined

- org.jfree.data.test.shiftTest.allValidParams(org.jfree.data.test.shiftTest) (1 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testNaNParam(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.getLowerBoundTest.testSmallValue(org.jfree.data.test.getLowerBoundTest) (0 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testNaNRange2(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.RangeTest (0 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testNoIssues(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.RangeCombineTest.testNullRange1(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.getLowerBoundTest.testBigValue(org.jfree.data.test.getLowerBoundTest) (0 ms)
- org.jfree.data.test.hashCodeTest.invalidRange(org.jfree.data.test.hashCodeTest) (0 ms)
- org.jfree.data.test.hashCodeTest.validRangeHash(org.jfree.data.test.hashCodeTest) (1 ms)
- org.jfree.data.test.RangeConstrainTest (0 ms)
- org.jfree.data.test.toStringTest.nullTest(org.jfree.data.test.toStringTest) (0 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.test1stNull(org.jfree.data.test.RangeCombineIgnoringNaN) (0 ms)
- org.jfree.data.test.hashCodeTest.twoNotEquals(org.jfree.data.test.hashCodeTest) (0 ms)
- org.jfree.data.test.RangeCombineTest.testLargeR2smallR1(org.jfree.data.test.RangeCombineTest) (0 ms)
- org.jfree.data.test.toStringTest.valids(org.jfree.data.test.toStringTest) (1 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.bothMinNaN(org.jfree.data.test.RangeCombineIgnoringNaN) (0 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.bothMaxNaN(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.RangeCombineTest.testBothRangesHaveNegativeUppers(org.jfree.data.test.RangeCombineTest) (0 ms)
- org.jfree.data.test.RangeCombineTest.testRangesBoundaryOverlap(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.increaseMutationTests.increaseMutationHashCode(org.jfree.data.test.increaseMutationTests) (1 ms)
- org.jfree.data.test.RangeGetLengthTest.testOneNegativeBound(org.jfree.data.test.RangeGetLengthTest) (1 ms)
- org.jfree.data.test.shiftTest.deltasZero(org.jfree.data.test.shiftTest) (0 ms)
- org.jfree.data.test.RangeEquals (0 ms)
- org.jfree.data.test.RangeTestShiftDouble (0 ms)
- org.jfree.data.test.RangeTestScale (1 ms)
- org.jfree.data.test.getLowerBoundTest.nullRange(org.jfree.data.test.getLowerBoundTest) (0 ms)
- org.jfree.data.test.RangeGetLengthTest.testLargePositive(org.jfree.data.test.RangeGetLengthTest) (0 ms)
- org.jfree.data.test.intersectsTesting.intersectNoOverlap(org.jfree.data.test.intersectsTesting) (0 ms)
- org.jfree.data.test.shiftTest.allowZeroFalse(org.jfree.data.test.shiftTest) (1 ms)
- org.jfree.data.test.RangeCombineTest.testRangesFullyNegative(org.jfree.data.test.RangeCombineTest) (0 ms)
- org.jfree.data.test.toStringTest.invalidUpper(org.jfree.data.test.toStringTest) (1 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testBothNaN(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.RangeCombineTest.testRangesPartiallyOverlap(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.intersectsTesting.intersectEqualRanges(org.jfree.data.test.intersectsTesting) (0 ms)
- org.jfree.data.test.shiftTest.allowZeroFalseWithZeroDelta(org.jfree.data.test.shiftTest) (1 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testRange2LowerisMin(org.jfree.data.test.RangeCombineIgnoringNaN) (0 ms)
- org.jfree.data.test.RangeGetLengthTest.testBothBoundariesZero(org.jfree.data.test.RangeGetLengthTest) (1 ms)
- org.jfree.data.test.getLowerBoundTest.testPositiveBoth(org.jfree.data.test.getLowerBoundTest) (0 ms)
- org.jfree.data.test.intersectsTesting.intersectInvalidRange(org.jfree.data.test.intersectsTesting) (0 ms)

- org.jfree.data.test.RangeCombineTest.testBothRangesHaveNegativeLowerers (org.jfree.data.test.RangeCombineTest) (0 ms)
- org.jfree.data.test.RangeGetLengthTest.testBothNegative(org.jfree.data.test.RangeGetLengthTest) (0 ms)
- org.jfree.data.test.intersectsTesting.intersectOverlapRanges(org.jfree.data.test.intersectsTesting) (1 ms)
- org.jfree.data.test.getLowerBoundTest.testZero(org.jfree.data.test.getLowerBoundTest) (1 ms)
- org.jfree.data.test.RangeGetLengthTest.testLowerBoundaryGreaterThanUpper (org.jfree.data.test.RangeGetLengthTest) (2 ms)
- org.jfree.data.test.shiftTest.nullTest(org.jfree.data.test.shiftTest) (1 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testNaNRange1(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.RangeTestExpandToInclude (2 ms)
- org.jfree.data.test.RangeGetLengthTest.testLowerBoundaryLessThanUpper (org.jfree.data.test.RangeGetLengthTest) (0 ms)
- org.jfree.data.test.getLowerBoundTest.flipFlop(org.jfree.data.test.getLowerBoundTest) (0 ms)
- org.jfree.data.test.shiftTest.deltasPositive(org.jfree.data.test.shiftTest) (1 ms)
- org.jfree.data.test.RangeCombineTest.testNullRange2(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.testBothNull(org.jfree.data.test.RangeCombineIgnoringNaN) (0 ms)
- org.jfree.data.test.RangeGetLengthTest.testLowerBoundaryEqualsUpperBoundary (org.jfree.data.test.RangeGetLengthTest) (1 ms)
- org.jfree.data.test.toStringTest.invalidLower(org.jfree.data.test.toStringTest) (0 ms)
- org.jfree.data.test.nanTest.invalidLowerBound(org.jfree.data.test.nanTest) (0 ms)
- org.jfree.data.test.hashCodeTest.twoEquals(org.jfree.data.test.hashCodeTest) (1 ms)
- org.jfree.data.test.nanTest.nullTest(org.jfree.data.test.nanTest) (0 ms)
- org.jfree.data.test.RangeCombineIgnoringNaN.test2ndNull(org.jfree.data.test.RangeCombineIgnoringNaN) (1 ms)
- org.jfree.data.test.hashCodeTest.nullRange(org.jfree.data.test.hashCodeTest) (1 ms)
- org.jfree.data.test.RangeExpand (0 ms)
- org.jfree.data.test.nanTest.invalidUpperBound(org.jfree.data.test.nanTest) (0 ms)
- org.jfree.data.test.RangeCombineTest.testNullBoth(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.RangeCombineTest.testRangesFullyOverlap(org.jfree.data.test.RangeCombineTest) (0 ms)
- org.jfree.data.test.nanTest.invalidBoth(org.jfree.data.test.nanTest) (0 ms)
- org.jfree.data.test.RangeTestIntersectsDouble (3 ms)
- org.jfree.data.test.intersectsTesting.intersectPartialOverlapRanges(org.jfree.data.test.intersectsTesting) (0 ms)
- org.jfree.data.test.RangeCombineTest.testRangesPartiallyNegative(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.shiftTest.allowZeroFalseWithPositiveDeltaTest(org.jfree.data.test.shiftTest) (0 ms)
- org.jfree.data.test.nanTest.valids(org.jfree.data.test.nanTest) (0 ms)
- org.jfree.data.test.getLowerBoundTest.testNegativeNumber(org.jfree.data.test.getLowerBoundTest) (0 ms)
- org.jfree.data.test.hashCodeTest.zeroRangeHash(org.jfree.data.test.hashCodeTest) (0 ms)
- org.jfree.data.test.RangeCombineTest.testRangesNoOverlap(org.jfree.data.test.RangeCombineTest) (1 ms)
- org.jfree.data.test.intersectsTesting.intersectNullRange(org.jfree.data.test.intersectsTesting) (1 ms)
- org.jfree.data.test.RangeGetLengthTest.testNull(org.jfree.data.test.RangeGetLengthTest) (0 ms)
- org.jfree.data.test.toStringTest.invalid(org.jfree.data.test.toStringTest) (0 ms)

Report generated by [PIT](#) 1.4.11