

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 <code>true</code> if the range contains the specified value and
137  * <code>false</code> otherwise.
138  *
139  * @param value the value to lookup.
140  *
141  * @return <code>true</code> 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 <code>true</code> if the range intersects with the specified
149  * range, and <code>false</code> 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 indicating whether the ranges intersect.
155  *
156  * @throws IllegalArgumentException if the lower bound is greater than the upper bound.
157  */
158 public boolean intersects(double b0, double b1) {
159     // Validate input bounds
160     19 if (b0 > b1) {
161         1 throw new IllegalArgumentException("Lower bound must be less than or equal to upper bound.");
162     }
163
164     // Check for intersection
165     49 boolean isDisjoint = b1 < this.lower || b0 > this.upper;
166     47 boolean isTouchingBoundary = b1 == this.lower || b0 == this.upper;
167
168     41 return !isDisjoint || isTouchingBoundary;
169 }
170
171 /**
172  * Returns <code>true</code> if the range intersects with the specified
173  * range, and <code>false</code> otherwise.
174  *
175  * @param range another range (<code>null</code> not permitted).
176  *
177  * @return A boolean.
178  *
179  * @since 1.0.9
180  */
181 public boolean intersects(Range range) {
182     6 return intersects(range.getLowerBound(), range.getUpperBound());
183 }
184
185 /**
186  * Returns the value within the range that is closest to the specified
187  * value.
188  *
189  * @param value the value.
190  *
191  * @return The constrained value.
192  */
193 public double constrain(double value) {
194     5 double result = value;
195     14 if (!contains(value)) {
196         19 if (value > this.upper) {
197             5 result = this.upper;
198         }
199         19 else if (value < this.lower) {
200             5 result = this.lower;
201         }
202     }

```

```

203 2         return result;
204     }
205
206     /**
207      * Creates a new range by combining two existing ranges.
208      * <P>
209      * Note that:
210      * <ul>
211      *   <li>either range can be <code>null</code>, in which case the other
212      *     range is returned;</li>
213      *   <li>if both ranges are <code>null</code> the return value is
214      *     <code>null</code>.</li>
215      * </ul>
216      *
217      * @param range1 the first range (<code>null</code> permitted).
218      * @param range2 the second range (<code>null</code> permitted).
219      *
220      * @return A new range (possibly <code>null</code>).
221      */
222     public static Range combine(Range range1, Range range2) {
223 4         if (range1 == null) {
224 2             return range2;
225         }
226 4         if (range2 == null) {
227 2             return range1;
228         }
229 4         double l = Math.min(range1.getLowerBound(), range2.getLowerBound());
230 4         double u = Math.max(range1.getUpperBound(), range2.getUpperBound());
231 13         return new Range(l, u);
232     }
233
234     /**
235      * Returns a new range that spans both <code>range1</code> and
236      * <code>range2</code>. This method has a special handling to ignore
237      * Double.NaN values.
238      *
239      * @param range1 the first range (<code>null</code> permitted).
240      * @param range2 the second range (<code>null</code> permitted).
241      *
242      * @return A new range (possibly <code>null</code>).
243      *
244      * @since 1.0.15
245      */
246     public static Range combineIgnoringNaN(Range range1, Range range2) {
247 4         if (range1 == null) {
248 13             if (range2 != null && range2.isNaNRange()) {
249 1                 return null;
250             }
251 2             return range2;
252         }
253 4         if (range2 == null) {
254 9             if (range1.isNaNRange()) {
255 1                 return null;
256             }
257 2             return range1;
258         }
259 4         double l = min(range1.getLowerBound(), range2.getLowerBound());
260 4         double u = max(range1.getUpperBound(), range2.getUpperBound());
261 28         if (Double.isNaN(l) && Double.isNaN(u)) {
262 1             return null;
263         }
264 13         return new Range(l, u);
265     }
266
267     /**
268      * Returns the minimum value. If either value is NaN, the other value is
269      * returned. If both are NaN, NaN is returned.
270      *
271      * @param d1 value 1.

```

```

272     * @param d2  value 2.
273     *
274     * @return The minimum of the two values.
275     */
276     private static double min(double d1, double d2) {
277         14 if (Double.isNaN(d1)) {
278             7 return d2;
279         }
280         14 if (Double.isNaN(d2)) {
281             7 return d1;
282         }
283         14 return Math.min(d1, d2);
284     }
285
286     private static double max(double d1, double d2) {
287         14 if (Double.isNaN(d1)) {
288             7 return d2;
289         }
290         14 if (Double.isNaN(d2)) {
291             7 return d1;
292         }
293         14 return Math.max(d1, d2);
294     }
295
296     /**
297     * Returns a range that includes all the values in the specified
298     * <code>range</code> AND the specified <code>value</code>.
299     *
300     * @param range  the range (<code>null</code> permitted).
301     * @param value  the value that must be included.
302     *
303     * @return A range.
304     *
305     * @since 1.0.1
306     */
307     public static Range expandToInclude(Range range, double value) {
308         4 if (range == null) {
309             13 return new Range(value, value);
310         }
311         15 if (value < range.getLowerBound()) {
312             9 return new Range(value, range.getUpperBound());
313         }
314         15 else if (value > range.getUpperBound()) {
315             9 return new Range(range.getLowerBound(), value);
316         }
317         else {
318             2 return range;
319         }
320     }
321
322     /**
323     * Creates a new range by adding margins to an existing range.
324     *
325     * @param range  the range (<code>null</code> not permitted).
326     * @param lowerMargin  the lower margin (expressed as a percentage of the
327     *                     range length).
328     * @param upperMargin  the upper margin (expressed as a percentage of the
329     *                     range length).
330     *
331     * @return The expanded range.
332     */
333     public static Range expand(Range range,
334                               double lowerMargin, double upperMargin) {
335         1 ParamChecks.checkNotNull(range, "range");
336         1 double length = range.getLength();
337         25 double lower = range.getLowerBound() - length * lowerMargin;
338         25 double upper = range.getUpperBound() + length * upperMargin;
339         19 if (lower > upper) {
340             // If the bounds cross, adjust them to the nearest valid values

```

```

341 8         lower = Math.min(lower, range.getUpperBound());
342 8         upper = Math.max(upper, range.getLowerBound());
343     }
344 13     return new Range(lower, upper);
345 }
346
347 /**
348  * Shifts the range by the specified amount.
349  *
350  * @param base the base range (<code>null</code> not permitted).
351  * @param delta the shift amount.
352  *
353  * @return A new range.
354  */
355 public static Range shift(Range base, double delta) {
356 14     return shift(base, delta, false);
357 }
358
359 /**
360  * Shifts the range by the specified amount.
361  *
362  * @param base the base range (<code>null</code> not permitted).
363  * @param delta the shift amount.
364  * @param allowZeroCrossing a flag that determines whether or not the
365  *                          bounds of the range are allowed to cross
366  *                          zero after adjustment.
367  *
368  * @return A new range.
369  */
370 public static Range shift(Range base, double delta,
371                          boolean allowZeroCrossing) {
372 1     ParamChecks.nullNotPermitted(base, "base");
373 13     if (allowZeroCrossing) {
374 16         return new Range(base.getLowerBound() + delta,
375 13                         base.getUpperBound() + delta);
376     }
377     else {
378 6         return new Range(shiftWithNoZeroCrossing(base.getLowerBound(),
379 8                 delta), shiftWithNoZeroCrossing(base.getUpperBound(),
380 5                 delta));
381     }
382 }
383
384 /**
385  * Returns the given <code>value</code> adjusted by <code>delta</code> but
386  * with a check to prevent the result from crossing <code>0.0</code>.
387  *
388  * @param value the value.
389  * @param delta the adjustment.
390  *
391  * @return The adjusted value.
392  */
393 private static double shiftWithNoZeroCrossing(double value, double delta) {
394 19     if (value > 0.0) {
395 26         return Math.max(value + delta, 0.0);
396     }
397     else if (value < 0.0) {
398 26         return Math.min(value + delta, 0.0);
399     }
400     else {
401 19         return value + delta;
402     }
403 }
404
405 /**
406  * Scales the range by the specified factor.
407  *
408  * @param base the base range (<code>null</code> not permitted).
409  * @param factor the scaling factor (must be non-negative).

```

```

410     *
411     * @return A new range.
412     *
413     * @since 1.0.9
414     */
415     public static Range scale(Range base, double factor) {
416 1       ParamChecks.nullNotPermitted(base, "base");
417 19      if (factor < 0) {
418 1          throw new IllegalArgumentException("Negative 'factor' argument.");
419      }
420 16      return new Range(base.getLowerBound() * factor,
421 13          base.getUpperBound() * factor);
422  }
423
424  /**
425   * Tests this object for equality with an arbitrary object.
426   *
427   * @param obj the object to test against (<code>null</code> permitted).
428   *
429   * @return A boolean.
430   */
431  @Override
432  public boolean equals(Object obj) {
433  3      if (this == obj) {
434  8          return true;
435      }
436  8      if (!(obj instanceof Range)) {
437  7          return false;
438      }
439
440      Range range = (Range) obj;
441
442  34      return Double.compare(this.lower, range.lower) == 0 &&
443  19          Double.compare(this.upper, range.upper) == 0;
444  }
445
446  /**
447   * Returns <code>true</code> if both the lower and upper bounds are
448   * <code>Double.NaN</code>, and <code>false</code> otherwise.
449   *
450   * @return A boolean.
451   *
452   * @since 1.0.18
453   */
454  public boolean isNaNRange() {
455  43      return Double.isNaN(this.lower) && Double.isNaN(this.upper);
456  }
457
458  /**
459   * Returns a hash code.
460   *
461   * @return A hash code.
462   */
463  @Override
464  public int hashCode() {
465      int result;
466      long temp;
467  6      temp = Double.doubleToLongBits(this.lower);
468  19      result = (int) (temp ^ (temp >>> 32));
469  6      temp = Double.doubleToLongBits(this.upper);
470  45      result = 29 * result + (int) (temp ^ (temp >>> 32));
471  7      return result;
472  }
473
474  /**
475   * Returns a string representation of this Range.
476   *
477   * @return A String "Range[lower,upper]" where lower=lower range and
478   *         upper=upper range.

```

```

479     */
480     @Override
481     public String toString() {
482 22         return ("Range[" + this.lower + "," + this.upper + "]");
483     }
484
485 }

```

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 (++) double local variable number 1 → KILLED
17. Incremented (++) 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 (++) 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 (++) 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 (++) 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 (++) double local variable number 3 → KILLED
6. Decrementd (--a) double local variable number 3 → KILLED
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
- 105 4. Incremented (a++) double field lower → KILLED
5. Decrementd (a--) double field lower → KILLED
6. Incremented (++) 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
3. Negated double field upper → KILLED
- 114 4. Incremented (a++) double field upper → KILLED
5. Decrementd (a--) double field upper → KILLED
6. Incremented (++) double field upper → KILLED
7. Decrementd (--a) double fieldupper → KILLED
- 123 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
 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
 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
- 132
1. replaced boolean return with false for org/jfree/data/Range::contains → KILLED
 2. replaced boolean return with true for org/jfree/data/Range::contains → 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 lower → KILLED
 17. Negated double local variable number 1 → KILLED
 18. Negated double field upper → 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
- 144

25. Substituted 0 with 1 → KILLED
 26. Substituted 1 with 0 → KILLED
 27. Substituted 0 with -1 → KILLED
 28. Less than to less or equal → KILLED
 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 → KILLED
 34. Less than to equal → KILLED
 35. greater than to equal → KILLED
 36. Less than to not equal → KILLED
 37. greater than to not equal → KILLED
 38. Incremented (a++) double local variable number 1 → KILLED
 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 → KILLED
 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 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
 160 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 → SURVIVED
 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 → SURVIVED
 18. Decrementd (--a) double local variable number 1 → KILLED
 19. Decrementd (--a) double local variable number 3 → KILLED
 161 1. removed call to java/lang/IllegalArgumentException::<init> → KILLED
 165 1. changed conditional boundary → SURVIVED
 2. changed conditional boundary → SURVIVED
 3. Substituted 0 with 1 → KILLED
 4. Substituted 1 with 0 → KILLED
 5. negated conditional → KILLED
 6. negated conditional → KILLED
 7. removed conditional - replaced comparison check with false → KILLED
 8. removed conditional - replaced comparison check with false → KILLED
 9. removed conditional - replaced comparison check with true → SURVIVED
 10. removed conditional - replaced comparison check with true → KILLED
 11. Negated double local variable number 3 → KILLED
 12. Negated double field lower → SURVIVED
 13. Negated double local variable number 1 → KILLED
 14. Negated double field upper → KILLED
 15. Substituted 0 with 1 → KILLED
 16. Substituted 1 with 0 → KILLED
 17. Substituted 0 with -1 → KILLED
 18. Substituted 1 with -1 → SURVIVED
 19. Substituted 1 with -1 → SURVIVED
 20. Substituted 0 with 1 → KILLED
 21. Substituted 1 with 2 → SURVIVED
 22. Substituted 0 with -1 → KILLED
 23. Substituted 1 with 0 → KILLED
 24. Less than to less or equal → SURVIVED
 25. greater than to less than → KILLED
 26. Less than to greater than → KILLED
 27. greater than to less or equal → KILLED
 28. Less than to greater or equal → KILLED
 29. greater than to greater or equal → SURVIVED
 30. Less than to equal → SURVIVED
 31. greater than to equal → KILLED
 32. Less than to not equal → KILLED
 33. greater than to not equal → KILLED
 34. Incremented (a++) double local variable number 3 → SURVIVED
 35. Incremented (a++) double field lower → SURVIVED
 36. Incremented (a++) double local variable number 1 → SURVIVED

37. Incremented (a++) double field upper → SURVIVED
 38. Decrementd (a--) double local variable number 3 → SURVIVED
 39. Decrementd (a--) double field lower → SURVIVED
 40. Decrementd (a--) double local variable number 1 → SURVIVED
 41. Decrementd (a--) double field upper → SURVIVED
 42. Incremented (a++) double local variable number 3 → SURVIVED
 43. Incremented (a++) double field lower → KILLED
 44. Incremented (a++) double local variable number 1 → KILLED
 45. Incremented (a++) double field upper → KILLED
 46. Decrementd (--a) double local variable number 3 → KILLED
 47. Decrementd (--a) double fieldlower → SURVIVED
 48. Decrementd (--a) double local variable number 1 → KILLED
 49. Decrementd (--a) double fieldupper → KILLED
 1. Substituted 0 with 1 → KILLED
 2. Substituted 1 with 0 → SURVIVED
 3. negated conditional → KILLED
 4. negated conditional → 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 → SURVIVED
 8. removed conditional - replaced equality check with true → SURVIVED
 9. Negated double local variable number 3 → SURVIVED
 10. Negated double field lower → SURVIVED
 11. Negated double local variable number 1 → SURVIVED
 12. Negated double field upper → SURVIVED
 13. Substituted 0 with 1 → KILLED
 14. Substituted 1 with 0 → SURVIVED
 15. Substituted 0 with -1 → KILLED
 16. Substituted 1 with -1 → SURVIVED
 17. Substituted 1 with -1 → SURVIVED
 18. Substituted 0 with 1 → KILLED
 19. Substituted 1 with 2 → SURVIVED
 20. Substituted 0 with -1 → KILLED
 21. Substituted 1 with 0 → SURVIVED
 22. equal to less than → SURVIVED
 23. equal to less than → SURVIVED
 24. equal to less or equal → SURVIVED
 25. equal to less or equal → SURVIVED
 26. equal to greater than → KILLED
 27. equal to greater than → KILLED
 28. equal to greater or equal → KILLED
 29. equal to greater or equal → KILLED
 30. equal to not equal → KILLED
 31. equal to not equal → KILLED
 32. Incremented (a++) double local variable number 3 → SURVIVED
 33. Incremented (a++) double field lower → SURVIVED
 34. Incremented (a++) double local variable number 1 → SURVIVED
 35. Incremented (a++) double field upper → SURVIVED
 36. Decrementd (a--) double local variable number 3 → SURVIVED
 37. Decrementd (a--) double field lower → SURVIVED
 38. Decrementd (a--) double local variable number 1 → SURVIVED
 39. Decrementd (a--) double field upper → SURVIVED
 40. Incremented (a++) double local variable number 3 → SURVIVED
 41. Incremented (a++) double field lower → SURVIVED
 42. Incremented (a++) double local variable number 1 → SURVIVED
 43. Incremented (a++) double field upper → SURVIVED
 44. Decrementd (--a) double local variable number 3 → SURVIVED
 45. Decrementd (--a) double fieldlower → SURVIVED
 46. Decrementd (--a) double local variable number 1 → SURVIVED
 47. Decrementd (--a) double fieldupper → SURVIVED
 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. Substituted 0 with 1 → KILLED
 4. Substituted 1 with 0 → KILLED
 5. negated conditional → KILLED
 6. negated conditional → KILLED
 7. removed conditional - replaced equality check with false → KILLED
 8. removed conditional - replaced equality check with false → KILLED
 9. removed conditional - replaced equality check with true → KILLED
 10. removed conditional - replaced equality check with true → SURVIVED
 11. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
 12. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
 13. Negated integer local variable number 5 → SURVIVED
 14. Negated integer local variable number 6 → SURVIVED
 15. Substituted 0 with 1 → KILLED
 16. Substituted 1 with 0 → KILLED
 17. Substituted 0 with -1 → KILLED
 18. Substituted 1 with -1 → SURVIVED
 19. Substituted 1 with -1 → SURVIVED
 20. Substituted 0 with 1 → KILLED
 21. Substituted 1 with 2 → KILLED
 22. Substituted 0 with -1 → KILLED
 23. Substituted 1 with 0 → KILLED
 24. equal to less than → KILLED
 25. not equal to less than → SURVIVED

26. equal to less or equal → SURVIVED
 27. not equal to less or equal → KILLED
 28. equal to greater than → KILLED
 29. not equal to greater than → SURVIVED
 30. equal to greater or equal → KILLED
 31. not equal to greater or equal → KILLED
 32. equal to not equal → KILLED
 33. not equal to equal → KILLED
 34. Incremented (a++) integer local variable number 5 → SURVIVED
 35. Incremented (a++) integer local variable number 6 → SURVIVED
 36. Decrementd (a--) integer local variable number 5 → SURVIVED
 37. Decrementd (a--) integer local variable number 6 → SURVIVED
 38. Incremented (++) integer local variable number 5 → KILLED
 39. Incremented (++) integer local variable number 6 → KILLED
 40. Decrementd (--) integer local variable number 5 → KILLED
 41. Decrementd (--) integer local variable number 6 → 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
 182 3. removed call to org/jfree/data/Range::getLowerBound → KILLED
 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 → SURVIVED
 194 3. Decrementd (a--) double local variable number 1 → SURVIVED
 4. Incremented (++) double local variable number 1 → KILLED
 5. Decrementd (--) double local variable number 1 → KILLED
 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 → SURVIVED
 6. not equal to less than → SURVIVED
 195 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 → SURVIVED
 12. Decrementd (a--) double local variable number 1 → SURVIVED
 13. Incremented (++) double local variable number 1 → SURVIVED
 14. Decrementd (--) double local variable number 1 → SURVIVED
 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
 196 10. Less or equal to equal → KILLED
 11. Less or equal to not equal → KILLED
 12. Incremented (a++) double local variable number 1 → SURVIVED
 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 (++) double local variable number 1 → SURVIVED
 17. Incremented (++) double field upper → KILLED
 18. Decrementd (--) double local variable number 1 → SURVIVED
 19. Decrementd (--) double fieldupper → KILLED
 1. Negated double field upper → KILLED
 2. Incremented (a++) double field upper → SURVIVED
 197 3. Decrementd (a--) double field upper → SURVIVED
 4. Incremented (++) double field upper → KILLED
 5. Decrementd (--) 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 → SURVIVED
 6. Negated double field lower → KILLED
 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
 199 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 (++) double local variable number 1 → SURVIVED
 17. Incremented (++) double field lower → KILLED
 18. Decrementd (--) double local variable number 1 → SURVIVED
 19. Decrementd (--) double fieldlower → KILLED

- [200](#) 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
- 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
- [203](#) 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
- [223](#) 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
- [224](#) 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
- [226](#) 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
- [227](#) 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
- [229](#) 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 → SURVIVED
- [230](#) 2. removed call to org/jfree/data/Range::getUpperBound → SURVIVED
- 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
- [231](#) 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
- [247](#) 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
- [248](#) 7. removed conditional - replaced equality check with true → SURVIVED
- 8. equal to less than → SURVIVED
- 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
- [249](#) 1. replaced return value with null for org/jfree/data/Range::combineIgnoringNaN → SURVIVED
- [251](#) 2. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to (if (x != null) null else throw new RuntimeException) → KILLED
- 1. negated conditional → KILLED
- [253](#) 2. removed conditional - replaced equality check with false → KILLED
- 3. removed conditional - replaced equality check with true → KILLED
- 4. not equal to equal → KILLED
- [254](#) 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
- 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

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

[257](#) 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

[259](#) 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

[260](#) 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 → SURVIVED
8. removed conditional - replaced equality check with true → SURVIVED
9. Negated double local variable number 2 → SURVIVED
10. Negated double local variable number 4 → SURVIVED
11. equal to less than → SURVIVED
12. equal to less than → SURVIVED
13. equal to less or equal → SURVIVED
[261](#) 14. equal to less or equal → SURVIVED
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 → SURVIVED
23. Decrementd (a--) double local variable number 2 → KILLED
24. Decrementd (a--) double local variable number 4 → SURVIVED
25. Incremented (++) double local variable number 2 → KILLED
26. Incremented (++) double local variable number 4 → SURVIVED
27. Decrementd (--a) double local variable number 2 → KILLED
28. Decrementd (--a) double local variable number 4 → SURVIVED

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

[264](#) 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 (++) double local variable number 2 → KILLED
11. Incremented (++) 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 → 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
[277](#) 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 (++) 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 → KILLED
[278](#) 4. Incremented (a++) double local variable number 2 → SURVIVED
5. Decrementd (a--) double local variable number 2 → SURVIVED
6. Incremented (++) double local variable number 2 → KILLED
7. Decrementd (--a) double local variable number 2 → KILLED

[280](#) 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 → SURVIVED

5. Negated double local variable number 2 → SURVIVED
 6. equal to less than → SURVIVED
 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 → SURVIVED
 12. Decrementd (a--) double local variable number 2 → KILLED
 13. Incremented (++a) double local variable number 2 → SURVIVED
 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
 281 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
 283 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 → SURVIVED
 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
 287 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
 288 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 → 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
 290 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::max → KILLED
 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → KILLED
 3. Negated double local variable number 0 → KILLED
 291 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
 293 1. replaced call to java/lang/Math::max with argument → SURVIVED
 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 → SURVIVED
 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 → SURVIVED

12. Incremented (++a) double local variable number 2 → KILLED
 13. Decrementd (--a) double local variable number 0 → SURVIVED
 14. Decrementd (--a) double local variable number 2 → KILLED

308 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

309 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 (++a) double local variable number 1 → KILLED
 11. Incremented (++a) 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
 7. greater or equal to less than → KILLED

311 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 (++a) 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

312 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. 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
 7. Less or equal to less than → SURVIVED

314 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 (++a) 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::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

315 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
 2. mutated return of Object value for org/jfree/data/Range::expandToInclude to (if (x != null) null else throw new RuntimeException) → KILLED

318

335 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → SURVIVED

336 1. removed call to org/jfree/data/Range::getLength → KILLED

337 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
 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
 338 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
 1. changed conditional boundary → SURVIVED
 2. negated conditional → SURVIVED
 3. removed conditional - replaced comparison check with false → SURVIVED
 4. removed conditional - replaced comparison check with true → SURVIVED
 5. Negated double local variable number 7 → SURVIVED
 6. Negated double local variable number 9 → SURVIVED
 7. Less or equal to less than → SURVIVED
 8. Less or equal to greater than → SURVIVED
 9. Less or equal to greater or equal → SURVIVED
 339 10. Less or equal to equal → SURVIVED
 11. Less or equal to not equal → SURVIVED
 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. replaced call to java/lang/Math::min with argument → NO_COVERAGE
 2. removed call to org/jfree/data/Range::getUpperBound → NO_COVERAGE
 3. removed call to java/lang/Math::min → NO_COVERAGE
 341 4. Negated double local variable number 7 → NO_COVERAGE
 5. Incremented (a++) double local variable number 7 → NO_COVERAGE
 6. Decrementd (a--) double local variable number 7 → NO_COVERAGE
 7. Incremented (++a) double local variable number 7 → NO_COVERAGE
 8. Decrementd (--a) double local variable number 7 → NO_COVERAGE
 1. replaced call to java/lang/Math::max with argument → NO_COVERAGE
 2. removed call to org/jfree/data/Range::getLowerBound → NO_COVERAGE
 3. removed call to java/lang/Math::max → NO_COVERAGE
 342 4. Negated double local variable number 9 → NO_COVERAGE
 5. Incremented (a++) double local variable number 9 → NO_COVERAGE
 6. Decrementd (a--) double local variable number 9 → NO_COVERAGE
 7. Incremented (++a) double local variable number 9 → NO_COVERAGE
 8. Decrementd (--a) double local variable number 9 → NO_COVERAGE
 344 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 (++a) double local variable number 7 → KILLED
11. Incremented (++a) 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 → SURVIVED
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
356 6. Negated double local variable number 1 → KILLED
7. Substituted 0 with 1 → SURVIVED
8. Substituted 0 with -1 → SURVIVED
9. Substituted 0 with 1 → SURVIVED
10. Substituted 0 with -1 → SURVIVED
11. Incremented (++a) double local variable number 1 → SURVIVED
12. Decrementd (a--) double local variable number 1 → SURVIVED
13. Incremented (++a) double local variable number 1 → KILLED
14. Decrementd (--a) double local variable number 1 → KILLED
372 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → SURVIVED
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
373 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 (++a) 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
374 7. Replaced double operation with first member → KILLED
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 (++a) double local variable number 1 → KILLED
16. Decrementd (--a) double local variable number 1 → KILLED
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
375 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
3. removed call to org/jfree/data/Range::getLowerBound → KILLED
376 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
3. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
377 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
378 1. Negated double local variable number 1 → KILLED
2. Incremented (++a) double local variable number 1 → SURVIVED
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 → 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 0 → 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
 394 10. Substituted 0.0 with -1.0 → KILLED
 11. Less or equal to less than → KILLED
 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
 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
 395 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 → 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 0 → 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
 397 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 → 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
 398 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
 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. Replaced double addition with subtraction → KILLED
 2. replaced double return with 0.0d for org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
 3. replaced return of double value with -(x + 1) for org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
 4. Negated double local variable number 0 → SURVIVED
 5. Negated double local variable number 2 → KILLED
 6. Replaced double operation with first member → KILLED
 7. Replaced double operation by second member → SURVIVED
 8. Replaced double addition with subtraction → KILLED
 9. Replaced double addition with multiplication → KILLED
 10. Replaced double addition with division → KILLED
 11. Replaced double addition with modulus → KILLED
 12. Incremented (a++) double local variable number 0 → SURVIVED
 13. Incremented (a++) double local variable number 2 → SURVIVED
 14. Decrementd (a--) double local variable number 0 → SURVIVED
 15. Decrementd (a--) double local variable number 2 → SURVIVED
 16. Incremented (a++) double local variable number 0 → KILLED
 17. Incremented (a++) double local variable number 2 → KILLED
 18. Decrementd (a--) double local variable number 0 → KILLED
 19. Decrementd (a--) double local variable number 2 → KILLED
 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 → SURVIVED
 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 → SURVIVED
 9. Substituted 0.0 with 1.0 → KILLED
 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 → 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
 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
 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
 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
 1. negated conditional → KILLED
 2. removed conditional - replaced equality check with true → KILLED
 3. not equal to equal → KILLED
 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. negated conditional → KILLED
 2. removed conditional - replaced equality check with false → KILLED
 3. removed conditional - replaced equality check with true → KILLED
 436 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
 437 3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
 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. replaced boolean return with false for org/jfree/data/Range::equals → KILLED
 2. replaced boolean return with true for org/jfree/data/Range::equals → KILLED
 3. Substituted 1 with 0 → KILLED
 4. Substituted 0 with 1 → KILLED
 5. negated conditional → KILLED
 6. removed call to java/lang/Double::compare → SURVIVED
 7. removed conditional - replaced equality check with false → KILLED
 8. removed conditional - replaced equality check with true → SURVIVED
 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 field lower → 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
 16. Substituted 0 with -1 → KILLED
 442 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. not equal to less than → SURVIVED
 23. not equal to less or equal → KILLED
 24. not equal to greater than → SURVIVED
 25. not equal to greater or equal → KILLED
 26. not equal to equal → KILLED
 27. Incremented (a++) double field lower → SURVIVED
 28. Incremented (a++) double field lower → SURVIVED
 29. Decrementd (a--) double field lower → SURVIVED
 30. Decrementd (a--) double field lower → SURVIVED
 31. Incremented (++a) double field lower → KILLED
 32. Incremented (++a) double field lower → KILLED
 33. Decrementd (--a) double fieldlower → KILLED
 34. Decrementd (--a) double fieldlower → KILLED
 1. negated conditional → KILLED
 2. removed call to java/lang/Double::compare → SURVIVED
 3. removed conditional - replaced equality check with false → KILLED
 4. removed conditional - replaced equality check with true → SURVIVED
 5. Negated double field upper → KILLED
 6. Negated double field upper → KILLED
 7. not equal to less than → SURVIVED
 8. not equal to less or equal → KILLED
 9. not equal to greater than → SURVIVED
 443 10. not equal to greater or equal → KILLED
 11. not equal to equal → KILLED
 12. Incremented (a++) double field upper → SURVIVED
 13. Incremented (a++) double field upper → SURVIVED
 14. Decrementd (a--) double field upper → SURVIVED
 15. Decrementd (a--) double field upper → SURVIVED
 16. Incremented (++a) double field upper → KILLED
 17. Incremented (++a) double field upper → KILLED
 18. Decrementd (--a) double fieldupper → KILLED
 19. Decrementd (--a) double fieldupper → KILLED
 455 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
 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 → SURVIVED
 467 3. Incremented (a++) double field lower → KILLED
 4. Decrementd (a--) double field lower → KILLED
 5. Incremented (++a) double field lower → KILLED
 6. Decrementd (--a) double fieldlower → KILLED
 1. Substituted 32 with 33 → SURVIVED
 2. Replaced Unsigned Shift Right with Shift Left → KILLED
 3. Replaced XOR with AND → KILLED
 4. Negated long local variable number 2 → SURVIVED
 5. Negated long local variable number 2 → SURVIVED
 6. Substituted 32 with 1 → KILLED
 7. Substituted 32 with 0 → KILLED
 8. Substituted 32 with -1 → KILLED
 9. Substituted 32 with -32 → SURVIVED
 468 10. Substituted 32 with 33 → SURVIVED
 11. Substituted 32 with 31 → SURVIVED
 12. Incremented (a++) long local variable number 2 → SURVIVED
 13. Incremented (a++) long local variable number 2 → SURVIVED
 14. Decrementd (a--) long local variable number 2 → SURVIVED
 15. Decrementd (a--) long local variable number 2 → SURVIVED
 16. Incremented (++a) long local variable number 2 → SURVIVED
 17. Incremented (++a) long local variable number 2 → SURVIVED
 18. Decrementd (--a) long local variable number 2 → SURVIVED
 19. Decrementd (--a) long local variable number 2 → SURVIVED
 1. removed call to java/lang/Double::doubleToLongBits → KILLED
 2. Negated double field upper → SURVIVED
 469 3. Incremented (a++) double field upper → KILLED
 4. Decrementd (a--) double field upper → KILLED
 5. Incremented (++a) double field upper → KILLED
 6. Decrementd (--a) double fieldupper → KILLED
 470 1. Substituted 29 with 30 → SURVIVED
 2. Substituted 32 with 33 → SURVIVED
 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 → SURVIVED
 7. Negated integer local variable number 1 → SURVIVED
 8. Negated long local variable number 2 → SURVIVED
 9. Negated long local variable number 2 → SURVIVED
 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 → SURVIVED
 16. Replaced integer multiplication with modulus → KILLED
 17. Replaced integer addition with multiplication → KILLED
 18. Replaced integer multiplication with addition → SURVIVED
 19. Replaced integer addition with division → KILLED
 20. Replaced integer multiplication with subtraction → SURVIVED
 21. Replaced integer addition with modulus → SURVIVED

```

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 → SURVIVED
27. Substituted 32 with -1 → KILLED
28. Substituted 29 with -29 → SURVIVED
29. Substituted 32 with -32 → SURVIVED
30. Substituted 29 with 30 → SURVIVED
31. Substituted 32 with 33 → SURVIVED
32. Substituted 29 with 28 → SURVIVED
33. Substituted 32 with 31 → SURVIVED
34. Incremented (a++) integer local variable number 1 → SURVIVED
35. Incremented (a++) long local variable number 2 → SURVIVED
36. Incremented (a++) long local variable number 2 → SURVIVED
37. Decrementd (a--) integer local variable number 1 → SURVIVED
38. Decrementd (a--) long local variable number 2 → SURVIVED
39. Decrementd (a--) long local variable number 2 → SURVIVED
40. Incremented (++) integer local variable number 1 → SURVIVED
41. Incremented (++) long local variable number 2 → SURVIVED
42. Incremented (++) long local variable number 2 → SURVIVED
43. Decrementd (--) integer local variable number 1 → SURVIVED
44. Decrementd (--) long local variable number 2 → SURVIVED
45. Decrementd (--) long local variable number 2 → SURVIVED
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 1 → SURVIVED
471 4. Incremented (a++) integer local variable number 1 → SURVIVED
5. Decrementd (a--) integer local variable number 1 → SURVIVED
6. Incremented (++) integer local variable number 1 → SURVIVED
7. Decrementd (--) integer local variable number 1 → SURVIVED
1. removed call to java/lang/StringBuilder::<init> → NO_COVERAGE
2. replaced return value with "" for org/jfree/data/Range::toString → NO_COVERAGE
3. removed call to java/lang/StringBuilder::append → NO_COVERAGE
4. removed call to java/lang/StringBuilder::append → NO_COVERAGE
5. removed call to java/lang/StringBuilder::append → NO_COVERAGE
6. removed call to java/lang/StringBuilder::append → NO_COVERAGE
7. removed call to java/lang/StringBuilder::toString → NO_COVERAGE
8. mutated return of Object value for org/jfree/data/Range::toString to ( if (x != null) null else throw
new RuntimeException ) → NO_COVERAGE
9. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
10. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
482 11. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
12. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
13. Negated double field lower → NO_COVERAGE
14. Negated double field upper → NO_COVERAGE
15. Incremented (a++) double field lower → NO_COVERAGE
16. Incremented (a++) double field upper → NO_COVERAGE
17. Decrementd (a--) double field lower → NO_COVERAGE
18. Decrementd (a--) double field upper → NO_COVERAGE
19. Incremented (++) double field lower → NO_COVERAGE
20. Incremented (++) double field upper → NO_COVERAGE
21. Decrementd (--) double fieldlower → NO_COVERAGE
22. Decrementd (--) double fieldupper → NO_COVERAGE

```

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.RangeTest (33 ms)

Report generated by [PIT](#) 1.6.8