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
      * You should have received a copy of the GNU Lesser General Public
19
      * License along with this library; if not, write to the Free Software
20
21
      * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
      * USA.
22
23
      * [Oracle and Java are registered trademarks of Oracle and/or its affiliates.
24
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
      package org.jfree.data;
64
65
66
      import java.io.Serializable;
67
      import org.jfree.chart.util.ParamChecks;
68
69
      /**
       * Represents an immutable range of values.
70
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
          /** The upper bound of the range. */
80
81
          private double upper;
82
          /**
83
84
           * Creates a new range.
85
           * @param lower the lower bound (must be <= upper bound).
86
87
           * @param upper the upper bound (must be >= lower bound).
88
89
          public Range(double lower, double upper) {
90 19
              if (lower > upper) {
91 <u>9</u>
                  String msg = "Range(double, double): require lower (" + lower
                      + ") <= upper (" + upper + ").";
92 11
93 <u>1</u>
                  throw new IllegalArgumentException(msg);
94
              }
95 <u>6</u>
              this.lower = lower;
              this.upper = upper;
96 6
97
          }
98
          /**
99
100
           * Returns the lower bound for the range.
101
102
           * @return The lower bound.
```

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

```
156
          public boolean intersects(double b0, double b1) {
              if (b0 <= this.lower) {</pre>
157 19
158 34
                  return (b1 > this.lower);
159
              }
160
              else {
                  return (b0 < this.upper && b1 >= b0);
161 53
              }
162
          }
163
164
165
          /**
166
           * Returns <code>true</code> if the range intersects with the specified
           * range, and <code>false</code> otherwise.
167
168
           * @param range another range (<code>null</code> not permitted).
169
170
171
           * @return A boolean.
172
           * @since 1.0.9
173
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
           * @param value the value.
183
184
           * @return The constrained value.
185
186
          public double constrain(double value) {
187
188 <del>5</del>
              double result = value;
              if (!contains(value)) {
189 14
                  if (value > this.upper) {
190 <del>19</del>
                       result = this.upper;
191 5
192
193 19
                  else if (value < this.lower) {</pre>
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
           * 
205
               <ii>>either range can be <code>null</code>, in which case the other
206
                    range is returned;
207
               if both ranges are <code>null</code> the return value is
208
                   <code>null</code>.
```

```
* 
209
210
           * @param range1 the first range (<code>null</code> permitted).
211
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 <sup>2</sup>
                   return range2;
219
               }
220 <u>4</u>
               if (range2 == null) {
221 <sup>2</sup>
                   return range1;
222
               double 1 = Math.min(range1.getLowerBound(), range2.getLowerBound());
223 4
224 4
               double u = Math.max(range1.getUpperBound()), range2.getUpperBound());
               return new Range(1, u);
225 13
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
           * @return A new range (possibly <code>null</code>).
236
237
           * @since 1.0.15
238
239
          public static Range combineIgnoringNaN(Range range1, Range range2) {
240
241 4
               if (range1 == null) {
                   if (range2 != null && range2.isNaNRange()) {
242 13
243 1
                       return null;
244
245 <sup>2</sup>
                   return range2;
246
               }
247 4
               if (range2 == null) {
                   if (range1.isNaNRange()) {
248 9
249 <u>1</u>
                       return null;
250
                   }
251 <sup>2</sup>
                   return range1;
252
253 4
               double 1 = min(range1.getLowerBound(), range2.getLowerBound());
254 <u>4</u>
               double u = max(range1.getUpperBound(), range2.getUpperBound());
               if (Double.isNaN(1) && Double.isNaN(u)) {
255 28
256 <u>1</u>
                   return null;
257
258 <u>13</u>
               return new Range(1, 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
           * @param d1 value 1.
265
            * @param d2 value 2.
266
267
268
           * @return The minimum of the two values.
269
          private static double min(double d1, double d2) {
270
271 14
               if (Double.isNaN(d1)) {
272 <u>7</u>
                   return d2;
273
               if (Double.isNaN(d2)) {
274 14
275 <u>7</u>
                   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
               if (Double.isNaN(d2)) {
284 <u>14</u>
285 7
                   return d1;
286
               }
287 14
              return Math.max(d1, d2);
288
          }
289
          /**
290
           * Returns a range that includes all the values in the specified
291
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
          public static Range expandToInclude(Range range, double value) {
301
302 4
               if (range == null) {
303 13
                   return new Range(value, value);
304
              if (value < range.getLowerBound()) {</pre>
305 15
306 9
                   return new Range(value, range.getUpperBound());
307
308 15
              else if (value > range.getUpperBound()) {
                   return new Range(range.getLowerBound(), value);
309 <u>9</u>
310
              }
              else {
311
312 <sup>2</sup>
                   return range;
               }
313
314
          }
```

```
315
          /**
316
           * Creates a new range by adding margins to an existing range.
317
318
           * @param range the range (<code>null</code> not permitted).
319
           * @param lowerMargin the lower margin (expressed as a percentage of the
320
321
                                  range length).
322
           * @param upperMargin the upper margin (expressed as a percentage of the
323
                                  range length).
324
           * @return The expanded range.
325
           */
326
327
          public static Range expand(Range range,
328
                                      double lowerMargin, double upperMargin) {
              ParamChecks.nullNotPermitted(range, "range");
329 1
330 1
              double length = range.getLength();
331 <u>25</u>
              double lower = range.getLowerBound() - length * lowerMargin;
              double upper = range.getUpperBound() + length * upperMargin;
332 25
333 19
              if (lower > upper) {
334 45
                  lower = lower / 2.0 + upper / 2.0;
335 5
                  upper = lower;
336
              return new Range(lower, upper);
337 13
338
          }
339
340
          /**
341
           * Shifts the range by the specified amount.
342
           * @param base the base range (<code>null</code> not permitted).
343
           * @param delta the shift amount.
344
345
           * @return A new range.
346
           */
347
348
          public static Range shift(Range base, double delta) {
              return shift(base, delta, false);
349 14
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");
              if (allowZeroCrossing) {
366 <u>13</u>
367 <u>16</u>
                  return new Range(base.getLowerBound() + delta,
```

```
368 13
                           base.getUpperBound() + delta);
369
              }
370
              else {
                   return new Range(shiftWithNoZeroCrossing(base.getLowerBound(),
371 <u>6</u>
3728
                           delta), shiftWithNoZeroCrossing(base.getUpperBound(),
                           delta));
373 5
374
              }
375
          }
376
377
          /**
378
           * Returns the given <code>value</code> adjusted by <code>delta</code> but
           * with a check to prevent the result from crossing <code>0.0</code>.
379
380
           * @param value the value.
381
           * @param delta the adjustment.
382
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
              else if (value < 0.0) {
390 19
391 <u>26</u>
                   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
           * @param base the base range (<code>null</code> not permitted).
401
           * @param factor the scaling factor (must be non-negative).
402
403
           * @return A new range.
404
405
406
           * @since 1.0.9
           */
407
408
          public static Range scale(Range base, double factor) {
409 1
              ParamChecks.nullNotPermitted(base, "base");
410 <del>19</del>
              if (factor < 0) {</pre>
                   throw new IllegalArgumentException("Negative 'factor' argument.");
411 1
412
              }
413 <u>16</u>
              return new Range(base.getLowerBound() * factor,
                       base.getUpperBound() * factor);
414 13
415
          }
416
          /**
417
418
           * Tests this object for equality with an arbitrary object.
419
           * @param obj the object to test against (<code>null</code> permitted).
420
```

```
421
422
            * @return A boolean.
           */
423
424
          @Override
425
          public boolean equals(Object obj) {
4268
               if (!(obj instanceof Range)) {
427 <u>7</u>
                   return false;
428
               }
429
              Range range = (Range) obj;
430 18
               if (!(this.lower == range.lower)) {
431 <u>7</u>
                   return false;
432
              if (!(this.upper == range.upper)) {
433 18
434 <u>7</u>
                   return false;
435
               }
4368
              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
           * @return A boolean.
443
444
445
           * @since 1.0.18
446
447
          public boolean isNaNRange() {
               return Double.isNaN(this.lower) && Double.isNaN(this.upper);
448 43
449
          }
450
          /**
451
452
           * Returns a hash code.
453
454
           * @return A hash code.
           */
455
456
          @Override
          public int hashCode() {
457
458
              int result;
459
               long temp;
              temp = Double.doubleToLongBits(this.lower);
460 6
461 <u>19</u>
               result = (int) (temp ^ (temp >>> 32));
462 <u>6</u>
              temp = Double.doubleToLongBits(this.upper);
               result = 29 * result + (int) (temp ^ (temp >>> 32));
463 45
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 <u>22</u>
               return ("Range[" + this.lower + "," + this.upper + "]");
476
477
478
      }
      Mutations

    changed conditional boundary → KILLED

      negated conditional → KILLED

    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
      Less or equal to less than → KILLED
      Less or equal to greater than → KILLED
      9. Less or equal to greater or equal → KILLED
90
      Less or equal to equal → KILLED
      11. Less or equal to not equal → KILLED
      12. Incremented (a++) double local variable number 1 \rightarrow KILLED 13. Incremented (a++) double local variable number 3 \rightarrow KILLED
      14. Decremented (a--) double local variable number 1 → KILLED
      15. Decremented (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. Decremented (--a) double local variable number 1 → KILLED
      19. Decremented (--a) double local variable number 3 → KILLED

    removed call to java/lang/StringBuilder::<init> → KILLED

    removed call to java/lang/StringBuilder::append → KILLED

    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

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

      7. Decremented (a--) double local variable number 1 → SURVIVED
      8. Incremented (++a) double local variable number 1 → SURVIVED
      9. Decremented (--a) double local variable number 1 → SURVIVED

    removed call to java/lang/StringBuilder::append → KILLED

    removed call to java/lang/StringBuilder::append → KILLED

      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

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

      9. Decremented (a--) double local variable number 3 → SURVIVED
      10. Incremented (++a) double local variable number 3 → SURVIVED 11. Decremented (--a) double local variable number 3 → SURVIVED
93

    removed call to java/lang/IllegalArgumentException::<init> → KILLED

      1. Removed assignment to member variable lower → KILLED

    Negated double local variable number 1 → KILLED

      3. Incremented (a++) double local variable number 1 → SURVIVED
95
      4. Decremented (a--) double local variable number 1 → SURVIVED

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

      6. Decremented (--a) double local variable number 1 \rightarrow KILLED
      1. Removed assignment to member variable upper → KILLED

    Negated double local variable number 3 → KILLED

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

96
      4. Decremented (a--) double local variable number 3 → SURVIVED
      5. Incremented (++a) double local variable number 3 → KILLED

    Decremented (--a) double local variable number 3 → KILLED

    replaced double return with 0.0d for org/jfree/data/Range::getLowerBound →

105
      KILLED
```

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

 Negated double field lower → KILLED

    Incremented (a++) double field lower → KILLED

    5. Decremented (a--) double field lower → KILLED
    6. Incremented (++a) double field lower → KILLED
    7. Decremented (--a) double fieldlower → KILLED

    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
      Negated double field upper → KILLED

4. Incremented (a++) double field upper → KILLED
5. Decremented (a--) double field upper → KILLED
6. Incremented (++a) double field upper → KILLED

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

    Replaced double subtraction with addition → KILLED

      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

    Replaced double operation by second member → KILLED

      8. Replaced double subtraction with addition → KILLED
123
      9. Replaced double subtraction with multiplication → KILLED

 Replaced double subtraction with division → KILLED

    Replaced double subtraction with modulus → KILLED

      12. Incremented (a++) double field upper → KILLED
      13. Incremented (a++) double field lower → KILLED
      14. Decremented (a--) double field upper → KILLED
      15. Decremented (a--) double field lower \rightarrow KILLED
      16. Incremented (++a) double field upper → KILLED 17. Incremented (++a) double field lower → KILLED 18. Decremented (--a) double fieldupper → KILLED
      19. Decremented (--a) double fieldlower → KILLED
      1. Substituted 2.0 with 1.0 → KILLED
132
      Substituted 2.0 with 1.0 → KILLED

    Replaced double division with multiplication → KILLED

    Replaced double division with multiplication → KILLED

      Replaced double addition with subtraction → KILLED
      6. replaced double return with 0.0d for org/jfree/data/Range::getCentralValue →
      KILLED
      7. replaced return of double value with -(x + 1) for
      org/jfree/data/Range::getCentralValue → KILLED
      8. Negated double field lower → KILLED
      Negated double field upper → KILLED

    Replaced double operation with first member → KILLED

    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 \rightarrow KILLED

    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

 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 → KILLED
```

```
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 → KILLED
      41. Incremented (a++) double field upper → KILLED
      42. Decremented (a--) double field lower → KILLED
      43. Decremented (a--) double field upper → KILLED
      44. Incremented (++a) double field lower → KILLED
      45. Incremented (++a) double field upper → KILLED
      46. Decremented (--a) double fieldlower → KILLED
      47. Decremented (--a) double fieldupper → KILLED
144

    replaced boolean return with false for org/jfree/data/Range::contains →

      KILLED
      replaced boolean return with true for org/jfree/data/Range::contains →
      NO_COVERAGE
      3. changed conditional boundary → SURVIVED

    changed conditional boundary → SURVIVED

      5. Substituted 1 with 0 → KILLED
      Substituted 0 with 1 → NO COVERAGE

 negated conditional → KILLED

      negated conditional → KILLED
      9. removed conditional - replaced comparison check with false \rightarrow KILLED
      10. removed conditional - replaced comparison check with false 
ightarrow KILLED
      11. removed conditional - replaced comparison check with true 
ightarrow SURVIVED
      12. removed conditional - replaced comparison check with true 
ightarrow 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) \rightarrow NO_COVERAGE
      15. Negated double local variable number 1 → KILLED

 Negated double field lower → KILLED

      17. Negated double local variable number 1 → KILLED
      18. Negated double field upper → KILLED
      19. Substituted 0 with 1 → NO_COVERAGE
      20. Substituted 1 with 0 → KILLED
      21. Substituted 1 with -1 → SURVIVED
      22. Substituted 0 with -1 → NO_COVERAGE
      23. Substituted 1 with -1 → SURVIVED
      24. Substituted 1 with 2 → KILLED
      25. Substituted 0 with 1 \rightarrow NO COVERAGE
      26. Substituted 1 with 0 → KILLED
      27. Substituted 0 with -1 → NO COVERAGE
      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 \rightarrow KILLED
      33. greater than to greater or equal → SURVIVED
      34. Less than to equal → SURVIVED
      35. greater than to equal → SURVIVED
      36. Less than to not equal → KILLED
      37. greater than to not equal → KILLED
      38. Incremented (a++) double local variable number 1 → SURVIVED
      39. Incremented (a++) double field lower → SURVIVED
      40. Incremented (a++) double local variable number 1 → SURVIVED
      41. Incremented (a++) double field upper → SURVIVED
      42. Decremented (a--) double local variable number 1 	o SURVIVED
      43. Decremented (a--) double field lower → SURVIVED
      44. Decremented (a--) double local variable number 1 	o SURVIVED
```

```
45. Decremented (a--) double field upper → SURVIVED
      46. Incremented (++a) double local variable number 1 → SURVIVED
      47. Incremented (++a) double field lower → SURVIVED
      48. Incremented (++a) double local variable number 1 → SURVIVED
       49. Incremented (++a) double field upper → SURVIVED
      50. Decremented (--a) double local variable number 1 → SURVIVED 51. Decremented (--a) double fieldlower → SURVIVED 52. Decremented (--a) double local variable number 1 → SURVIVED
       53. Decremented (--a) double fieldupper → SURVIVED

    changed conditional boundary → SURVIVED

       negated conditional → KILLED
       3. removed conditional - replaced comparison check with false \rightarrow KILLED

    removed conditional - replaced comparison check with true → SURVIVED

      5. Negated double local variable number 1 → SURVIVED6. 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 \rightarrow SURVIVED
      10. greater than to equal → SURVIVED
157

 greater than to not equal → KILLED

       12. Incremented (a++) double local variable number 1 \rightarrow SURVIVED
       13. Incremented (a++) double field lower → KILLED
       14. Decremented (a--) double local variable number 1 \rightarrow SURVIVED
       15. Decremented (a--) double field lower → SURVIVED
       16. Incremented (++a) double local variable number 1 → SURVIVED
       17. Incremented (++a) double field lower → KILLED

    Decremented (--a) double local variable number
    Decremented (--a) double fieldlower → SURVIVED

                          [--a) double local variable number 1 → SURVIVED

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

      KILLED
       replaced boolean return with true for org/jfree/data/Range::intersects →
      KILLED
       changed conditional boundary → SURVIVED
       4. Substituted 1 with 0 → KILLED
       Substituted 0 with 1 → KILLED
       6. negated conditional → KILLED
      7. removed conditional - replaced comparison check with false \rightarrow KILLED 8. removed conditional - replaced comparison check with true \rightarrow KILLED
      9. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow KILLED
       10. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow 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
       16. Substituted 0 with -1 → KILLED
158
       17. Substituted 1 with -1 \rightarrow 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 → SURVIVED
       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 → SURVIVED
       29. Decremented (a--) double local variable number 3 → SURVIVED
       30. Decremented (a--) double field lower → SURVIVED
       31. Incremented (++a) double local variable number 3 → SURVIVED
       32. Incremented (++a) double field lower → KILLED
      33. Decremented (--a) double local variable number 34. Decremented (--a) double fieldlower → SURVIVED
                          [--a) double local variable number 3 → KILLED
161

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

      NO COVERAGE
```

```
replaced boolean return with true for org/jfree/data/Range::intersects →
       NO_COVERAGE
       changed conditional boundary → NO_COVERAGE

    changed conditional boundary → NO_COVERAGE

       Substituted 1 with 0 → NO_COVERAGE
       Substituted 0 with 1 → NO_COVERAGE
       7. negated conditional → NO_COVERAGE
       8. negated conditional → NO_COVERAGE
       9. removed conditional - replaced comparison check with false → NO_COVERAGE
       10. removed conditional - replaced comparison check with false \rightarrow NO\_COVERAGE
       11. removed conditional - replaced comparison check with true \rightarrow NO_\overline{\text{C}}OVERAGE
       12. removed conditional - replaced comparison check with true → NO_COVERAGE
       13. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow NO_COVERAGE
       14. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow NO_{COVERAGE}
       15. Negated double local variable number 1 → NO_COVERAGE

 Negated double field upper → NO_COVERAGE

       17. Negated double local variable number 3 → NO_COVERAGE
       18. Negated double local variable number 1 → NO_COVERAGE
       19. Substituted 0 with 1 → NO_COVERAGE
       20. Substituted 1 with 0 → NO_COVERAGE
       21. Substituted 1 with -1 → NO_COVERAGE
       22. Substituted 0 with -1 → NO_COVERAGE
       23. Substituted 1 with -1 → NO_COVERAGE
       24. Substituted 1 with 2 → NO_COVERAGE
       25. Substituted 0 with 1 → NO_COVERAGE
       26. Substituted 1 with 0 → NO COVERAGE
       27. Substituted 0 with -1 → NO COVERAGE
       28. greater or equal to less than → NO_COVERAGE
       29. Less than to less or equal → NO_COVERAGE
       30. greater or equal to less or equal → NO_COVERAGE
       31. Less than to greater than → NO_COVERAGE
       32. greater or equal to greater than \rightarrow NO\_COVERAGE
       33. Less than to greater or equal → NO_COVERAGE
       34. greater or equal to equal → NO_COVERAGE
       35. Less than to equal → NO_COVERAGE
       36. greater or equal to not equal → NO_COVERAGE
       37. Less than to not equal → NO_COVERAGE
       38. Incremented (a++) double local variable number 1 → NO_COVERAGE 39. Incremented (a++) double field upper → NO_COVERAGE
       40. Incremented (a++) double local variable number 3 → NO_COVERAGE
       41. Incremented (a++) double local variable number 1 → NO_COVERAGE
       42. Decremented (a--) double local variable number 1 \rightarrow NO_{COVERAGE}
       43. Decremented (a--) double field upper → NO COVERAGE
       44. Decremented (a--) double local variable number 3 → NO_COVERAGE
       45. Decremented (a--) double local variable number 1 → NO_COVERAGE
       46. Incremented (++a) double local variable number 1 	o NO_{COVERAGE}
       47. Incremented (++a) double field upper → NO_COVERAGE
       48. Incremented (++a) double local variable number 3 → NO_COVERAGE
       49. Incremented (++a) double local variable number 1 → NO_COVERAGE
       50. Decremented (--a) double local variable number 1 → NO_COVERAGE 51. Decremented (--a) double fieldupper → NO_COVERAGE 52. Decremented (--a) double local variable number 3 → NO_COVERAGE
       53. Decremented (--a) double local variable number 1 → NO_COVERAGE

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

       SURVIVED
       replaced boolean return with true for org/jfree/data/Range::intersects →
       KILLED
176
      3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
4. removed call to org/jfree/data/Range::getUpperBound → SURVIVED
5. removed call to org/jfree/data/Range::intersects → SURVIVED
       6. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow KILLED

    Negated double local variable number 1 → KILLED

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

    Decremented (a--) double local variable number 1 → SURVIVED
    Incremented (++a) double local variable number 1 → KILLED
    Decremented (--a) double local variable number 1 → KILLED
```

```
189
       1. negated conditional → SURVIVED
       2. removed call to org/jfree/data/Range::contains → SURVIVED
       3. removed conditional - replaced equality check with false \rightarrow SURVIVED
       4. removed conditional - replaced equality check with true \rightarrow SURVIVED
       5. Negated double local variable number 1 → SURVIVED
       6. not equal to less than → SURVIVED

 not equal to less or equal → SURVIVED

       8. not equal to greater than → SURVIVED
       9. not equal to greater or equal → SURVIVED
       10. not equal to equal → SURVIVED
       11. Incremented (a++) double local variable number 1 \rightarrow SURVIVED
       12. Decremented (a--) double local variable number 1 \rightarrow SURVIVED
       13. Incremented (++a) double local variable number 1 \rightarrow SURVIVED
       14. Decremented (--a) double local variable number 1 \rightarrow SURVIVED

    changed conditional boundary → NO_COVERAGE

       negated conditional → NO_COVERAGE
       3. removed conditional - replaced comparison check with false → NO_COVERAGE
       4. removed conditional - replaced comparison check with true → NO_COVERAGE

    Negated double local variable number 1 → NO_COVERAGE

       6. Negated double field upper → NO COVERAGE
       7. Less or equal to less than → NO COVERAGE
       8. Less or equal to greater than → NO COVERAGE
       9. Less or equal to greater or equal → NO_COVERAGE
190
       10. Less or equal to equal → NO_COVERAGE
       11. Less or equal to not equal → NO_COVERAGE
       12. Incremented (a++) double local variable number 1 \rightarrow NO_{COVERAGE}
       13. Incremented (a++) double field upper → NO_COVERAGE
14. Decremented (a--) double local variable number 1 → NO_COVERAGE
15. Decremented (a--) double field upper → NO_COVERAGE
16. Incremented (++a) double local variable number 1 → NO_COVERAGE
       17. Incremented (++a) double field upper → NO_COVERAGE
       18. Decremented (--a) double local variable number 1 → NO_COVERAGE
       19. Decremented (--a) double fieldupper → NO_COVERAGE

    Negated double field upper → NO_COVERAGE

    Incremented (a++) double field upper → NO_COVERAGE
    Decremented (a--) double field upper → NO_COVERAGE
    Incremented (++a) double field upper → NO_COVERAGE
    Decremented (--a) double fieldupper → NO_COVERAGE

191

    changed conditional boundary → NO_COVERAGE

 negated conditional → NO_COVERAGE

       3. removed conditional - replaced comparison check with false \rightarrow NO_COVERAGE
       4. removed conditional - replaced comparison check with true \rightarrow NO_COVERAGE

    Negated double local variable number 1 → NO_COVERAGE

       6. Negated double field lower → NO_COVERAGE
       7. greater or equal to less than → NO_COVERAGE
       8. greater or equal to less or equal → NO_COVERAGE
       9. greater or equal to greater than → NO_COVERAGE
       10. greater or equal to equal → NO_COVERAGE
       11. greater or equal to not equal → NO_COVERAGE
       12. Incremented (a++) double local variable number 1 \rightarrow NO\_COVERAGE
       13. Incremented (a++) double field lower → NO_COVERAGE
       14. Decremented (a--) double local variable number 1 \rightarrow NO_{COVERAGE}
       15. Decremented (a--) double field lower → NO_COVERAGE
       16. Incremented (++a) double local variable number 1 \rightarrow NO_COVERAGE
       17. Incremented (++a) double field lower → NO_COVERAGE
18. Decremented (--a) double local variable number 1 → NO_COVERAGE
19. Decremented (--a) double fieldlower → NO_COVERAGE

    Negated double field lower → NO COVERAGE

       2. Incremented (a++) double field lower → NO_COVERAGE
<u>194</u>

    Decremented (a--) double field lower → NO_COVERAGE

    4. Incremented (++a) double field lower → NO_COVERAGE
    5. Decremented (--a) double fieldlower → NO_COVERAGE

    replaced double return with 0.0d for org/jfree/data/Range::constrain →

197
       KILLED
```

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

 Negated double local variable number 3 → KILLED

       4. Incremented (a++) double local variable number 3 → SURVIVED
       5. Decremented (a--) double local variable number 3 → SURVIVED 6. Incremented (++a) double local variable number 3 → KILLED 7. Decremented (--a) double local variable number 3 → KILLED

    negated conditional → KILLED

       removed conditional - replaced equality check with false → KILLED
217
       3. removed conditional - replaced equality check with true → KILLED
       4. not equal to equal → KILLED

    replaced return value with null for org/jfree/data/Range::combine → KILLED

218
       mutated return of Object value for org/jfree/data/Range::combine to ( if
       (x != null) null else throw new RuntimeException ) \rightarrow KILLED

    negated conditional → KILLED

       2. removed conditional - replaced equality check with false \rightarrow KILLED 3. removed conditional - replaced equality check with true \rightarrow SURVIVED
220
       4. not equal to equal → KILLED

    replaced return value with null for org/jfree/data/Range::combine → KILLED

       2. mutated return of Object value for org/jfree/data/Range::combine to ( if
221
       (x != null) null else throw new RuntimeException ) → KILLED

    replaced call to java/lang/Math::min with argument → NO_COVERAGE

       2. removed call to org/jfree/data/Range::getLowerBound → NO_COVERAGE
223
       3. removed call to org/jfree/data/Range::getLowerBound → NO_COVERAGE

 removed call to java/lang/Math::min → NO_COVERAGE

    replaced call to java/lang/Math::max with argument → NO_COVERAGE

       removed call to org/jfree/data/Range::getUpperBound → NO COVERAGE
224

    removed call to org/jfree/data/Range::getUpperBound → NO_COVERAGE

 removed call to java/lang/Math::max → NO_COVERAGE

    removed call to org/jfree/data/Range::<init> → NO_COVERAGE

       replaced return value with null for org/jfree/data/Range::combine →
       NO COVERAGE
       mutated return of Object value for org/jfree/data/Range::combine to ( if
       (x != null) null else throw new RuntimeException ) → NO COVERAGE

    Negated double local variable number 2 → NO_COVERAGE

       5. Negated double local variable number 4 → NO_COVERAGE
225
       6. Incremented (a++) double local variable number 2 → NO_COVERAGE
       7. Incremented (a++) double local variable number 4 → NO_COVERAGE 8. Decremented (a--) double local variable number 2 → NO_COVERAGE 9. Decremented (a--) double local variable number 4 → NO_COVERAGE
       10. Incremented (++a) double local variable number 2 \rightarrow NO\_COVERAGE 11. Incremented (++a) double local variable number 4 \rightarrow NO\_COVERAGE 12. Decremented (--a) double local variable number 2 \rightarrow NO\_COVERAGE 13. Decremented (--a) double local variable number 4 \rightarrow NO\_COVERAGE

    negated conditional → KILLED

       2. removed conditional - replaced equality check with false \rightarrow SURVIVED 3. removed conditional - replaced equality check with true \rightarrow KILLED
241
       not equal to equal → KILLED

    negated conditional → NO_COVERAGE

 negated conditional → NO_COVERAGE

       removed call to org/jfree/data/Range::isNaNRange → NO_COVERAGE

    removed conditional - replaced equality check with false → NO_COVERAGE

       5. removed conditional - replaced equality check with false → NO_COVERAGE
       6. removed conditional - replaced equality check with true \rightarrow NO_COVERAGE
       7. removed conditional - replaced equality check with true 
ightarrow NO_COVERAGE
       equal to less than → NO_COVERAGE

 equal to less or equal → NO_COVERAGE

       10. equal to greater than → NO_COVERAGE
       11. equal to greater or equal → NO_COVERAGE
       12. equal to not equal → NO_COVERAGE
       13. equal to not equal → NO_COVERAGE

    mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN

243
       to ( if (x != null) null else throw new RuntimeException ) \rightarrow NO_COVERAGE
245
```

```
    replaced return value with null for org/jfree/data/Range::combineIgnoringNaN

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

    negated conditional → KILLED

      removed conditional - replaced equality check with false → SURVIVED
247

    removed conditional - replaced equality check with true → KILLED

      4. not equal to equal → KILLED

    negated conditional → NO_COVERAGE

      removed call to org/jfree/data/Range::isNaNRange → NO_COVERAGE

    removed conditional - replaced equality check with false → NO_COVERAGE

    removed conditional - replaced equality check with true → NO_COVERAGE

248
      5. equal to less than → NO COVERAGE
      equal to less or equal → NO_COVERAGE
      7. equal to greater than → NO_COVERAGE
      equal to greater or equal → NO_COVERAGE
      equal to not equal → NO_COVERAGE

    mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN

249
      to ( if (x != null) null else throw new RuntimeException ) \rightarrow NO_COVERAGE

    replaced return value with null for org/jfree/data/Range::combineIgnoringNaN

      → NO_COVERAGE
251
      mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN
      to ( if (x != null) null else throw new RuntimeException ) \rightarrow NO_COVERAGE

    replaced call to org/jfree/data/Range::min with argument → KILLED

      2. removed call to org/jfree/data/Range::getLowerBound → KILLED
253
      3. removed call to org/jfree/data/Range::getLowerBound → KILLED
      4. removed call to org/jfree/data/Range::min → KILLED

    replaced call to org/jfree/data/Range::max with argument → SURVIVED

      2. removed call to org/jfree/data/Range::getUpperBound → KILLED
254
      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
      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 \rightarrow KILLED
      7. removed conditional - replaced equality check with true → SURVIVED
      8. removed conditional - replaced equality check with true \rightarrow SURVIVED
      9. Negated double local variable number 2 → SURVIVED

    Negated double local variable number 4 → SURVIVED

      equal to less than → SURVIVED
      12. equal to less than → SURVIVED
      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 → SURVIVED
      23. Decremented (a--) double local variable number 2 → KILLED
      24. Decremented (a--) double local variable number 4 → SURVIVED 25. Incremented (++a) double local variable number 2 → KILLED
      26. Incremented (++a) double local variable number 4 → SURVIVED 27. Decremented (--a) double local variable number 2 → KILLED
      28. Decremented (--a) double local variable number 4 → SURVIVED
      1. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN
256
      to ( if (x != null) null else throw new RuntimeException ) \rightarrow KILLED
258

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

      replaced return value with null for org/jfree/data/Range::combineIgnoringNaN
      → KILLED
```

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

    Negated double local variable number 2 → KILLED

 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. Decremented (a--) double local variable number 2 → SURVIVED 9. Decremented (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. Decremented (--a) double local variable number 2 → KILLED
      13. Decremented (--a) double local variable number 4 → KILLED

    negated conditional → KILLED

      removed call to java/lang/Double::isNaN → SURVIVED
      3. removed conditional - replaced equality check with false \rightarrow 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
271
      8. equal to greater than → KILLED
      equal to greater or equal → SURVIVED
      equal to not equal → KILLED
      11. Incremented (a++) double local variable number 0 → KILLED
      12. Decremented (a--) double local variable number 0 → KILLED
      13. Incremented (++a) double local variable number 0 → KILLED
      14. Decremented (--a) double local variable number 0 → KILLED

    replaced double return with 0.0d for org/jfree/data/Range::min → KILLED

      replaced return of double value with -(x + 1) for org/jfree/data/Range::min
      → KILLED
      3. Negated double local variable number 2 → SURVIVED
272
      4. Incremented (a++) double local variable number 2 \rightarrow SURVIVED
      5. Decremented (a--) double local variable number 2 → SURVIVED

    Incremented (++a) double local variable number 2 → SURVIVED

      7. Decremented (--a) double local variable number 2 → SURVIVED

    negated conditional → SURVIVED

      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 → SURVIVED
      5. Negated double local variable number 2 → SURVIVED
      6. equal to less than → SURVIVED

 equal to less or equal → SURVIVED

274
      equal to greater than → SURVIVED
      equal to greater or equal → SURVIVED
      10. equal to not equal → SURVIVED
      11. Incremented (a++) double local variable number 2 → SURVIVED
      12. Decremented (a--) double local variable number 2 → SURVIVED
      13. Incremented (++a) double local variable number 2 \rightarrow SURVIVED
      14. Decremented (--a) double local variable number 2 → SURVIVED

    replaced double return with 0.0d for org/jfree/data/Range::min → NO_COVERAGE

      replaced return of double value with -(x + 1) for org/jfree/data/Range::min
      → NO COVERAGE

 Negated double local variable number 0 → NO_COVERAGE

275

    Incremented (a++) double local variable number 0 → NO_COVERAGE

      5. Decremented (a--) double local variable number 0 → NO_COVERAGE
      6. Incremented (++a) double local variable number 0 → NO_COVERAGE
      7. Decremented (--a) double local variable number 0 → NO_COVERAGE

    replaced call to java/lang/Math::min with argument → KILLED
    removed call to java/lang/Math::min → KILLED

    replaced double return with 0.0d for org/jfree/data/Range::min → KILLED

      4. replaced return of double value with -(\bar{x} + 1) for org/jfree/data/Range::min
      → KILLED

    Negated double local variable number 0 → KILLED

 Negated double local variable number 2 → KILLED

    Incremented (a++) double local variable number 0 → SURVIVED

    Incremented (a++) double local variable number 2 → SURVIVED
```

```
9. Decremented (a--) double local variable number 0 → SURVIVED

    Decremented (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. Decremented (--a) double local variable number 0 → KILLED 14. Decremented (--a) double local variable number 2 → SURVIVED

    negated conditional → SURVIVED

       removed call to java/lang/Double::isNaN → SURVIVED

    removed conditional - replaced equality check with false → SURVIVED

       4. removed conditional - replaced equality check with true 
ightarrow SURVIVED
       5. Negated double local variable number 0 → SURVIVED
       equal to less than → SURVIVED

 equal to less or equal → SURVIVED

      8. equal to greater than → SURVIVED
      9. equal to greater or equal → SURVIVED
       10. equal to not equal → SURVIVED
       11. Incremented (a++) double local variable number 0 → SURVIVED
      12. Decremented (a--) double local variable number 0 → SURVIVED

    Incremented (++a) double local variable number 0 → SURVIVED

       14. Decremented (--a) double local variable number 0 → SURVIVED

    replaced double return with 0.0d for org/jfree/data/Range::max → KILLED

       replaced return of double value with -(x + 1) for org/jfree/data/Range::max
      → KILLED
      3. Negated double local variable number 2 → SURVIVED
282
      4. Incremented (a++) double local variable number 2 \rightarrow SURVIVED
       5. Decremented (a--) double local variable number 2 → SURVIVED
      6. Incremented (++a) double local variable number 2 → SURVIVED
      7. Decremented (--a) double local variable number 2 → SURVIVED

    negated conditional → KILLED

       2. removed call to java/lang/Double::isNaN → SURVIVED

    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
284
      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. Decremented (a--) double local variable number 2 → KILLED
      13. Incremented (++a) double local variable number 2 → KILLED 14. Decremented (--a) double local variable number 2 → KILLED

    replaced double return with 0.0d for org/jfree/data/Range::max → NO_COVERAGE

       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
285

    Incremented (a++) double local variable number 0 → NO_COVERAGE

       5. Decremented (a--) double local variable number 0 → NO_COVERAGE
      6. Incremented (++a) double local variable number 0 → NO_COVERAGE7. Decremented (--a) double local variable number 0 → NO_COVERAGE

    replaced call to java/lang/Math::max with argument → SURVIVED

       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 -(\bar{x} + 1) for org/jfree/data/Range::max
      → KILLED
      5. Negated double local variable number 0 \rightarrow SURVIVED 6. Negated double local variable number 2 \rightarrow KILLED
      7. Incremented (a++) double local variable number 0 \rightarrow SURVIVED 8. Incremented (a++) double local variable number 2 \rightarrow SURVIVED
287
      9. Decremented (a--) double local variable number 0 → SURVIVED
      10. Decremented (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. Decremented (--a) double local variable number 0 → SURVIVED
       14. Decremented (--a) double local variable number 2 → KILLED
```

```
302
       1. negated conditional → KILLED
       2. removed conditional - replaced equality check with false \rightarrow KILLED
       3. removed conditional - replaced equality check with true → KILLED
       not equal to equal → KILLED

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

       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 ) \rightarrow KILLED
       4. Negated double local variable number 1 → SURVIVED
       5. Negated double local variable number 1 → SURVIVED

    Incremented (a++) double local variable number 1 → KILLED
    Incremented (a++) double local variable number 1 → SURVIVED
    Decremented (a--) double local variable number 1 → KILLED
    Decremented (a--) double local variable number 1 → SURVIVED

303
       10. Incremented (++a) double local variable number 1 \rightarrow KILLED 11. Incremented (++a) double local variable number 1 \rightarrow KILLED
       12. Decremented (--a) double local variable number 1 → KILLED
       13. Decremented (--a) double local variable number 1 → KILLED

    changed conditional boundary → SURVIVED

       negated conditional → KILLED
       3. removed call to org/jfree/data/Range::getLowerBound → KILLED
       4. removed conditional - replaced comparison check with false \rightarrow KILLED 5. removed conditional - replaced comparison check with true \rightarrow KILLED
       6. Negated double local variable number 1 → 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
       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. Decremented (a--) double local variable number 1 → KILLED
       14. Incremented (++a) double local variable number 1 → KILLED 15. Decremented (--a) double local variable number 1 → KILLED

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

       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 ) \rightarrow KILLED
       5. Negated double local variable number 1 → SURVIVED

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

    7. Decremented (a--) double local variable number 1 → SURVIVED
    8. Incremented (++a) double local variable number 1 → KILLED
    9. Decremented (--a) double local variable number 1 → KILLED

    changed conditional boundary → SURVIVED

       negated conditional → KILLED
       3. removed call to org/jfree/data/Range::getUpperBound \rightarrow 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
       8. Less or equal to greater than → KILLED
       9. Less or equal to greater or equal → KILLED
       10. Less or equal to equal → KILLED

 Less or equal to not equal → KILLED

       12. Incremented (a++) double local variable number 1 	o 	ext{KILLED}
       13. Decremented (a--) double local variable number 1 → KILLED
       14. Incremented (++a) double local variable number 1 → KILLED
       15. Decremented (--a) double local variable number 1 → KILLED
309

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

       removed call to org/jfree/data/Range::getLowerBound → KILLED
       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 ) \rightarrow KILLED
      5. Negated double local variable number 1 → KILLED
      6. Incremented (a++) double local variable number 1 → SURVIVED
      7. Decremented (a--) double local variable number 1 → SURVIVED 8. Incremented (++a) double local variable number 1 → KILLED 9. Decremented (--a) double local variable number 1 → KILLED

    replaced return value with null for org/jfree/data/Range::expandToInclude →

      KILLED
312
      mutated return of Object value for org/jfree/data/Range::expandToInclude to
      ( if (x != null) null else throw new RuntimeException ) \rightarrow KILLED

    removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted →

329
      SURVIVED
330

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

    Replaced double multiplication with division → KILLED

      Replaced double subtraction with addition → KILLED
      3. removed call_to org/jfree/data/Range::getLowerBound → KILLED
      4. Negated double local variable number 5 → KILLED

 Negated double local variable number 1 → KILLED

      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

    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 \rightarrow KILLED 19. Incremented (a++) double local variable number 1 \rightarrow SURVIVED
      20. Decremented (a--) double local variable number 5 \rightarrow \text{KILLED} 21. Decremented (a--) double local variable number 1 \rightarrow \text{SURVIVED} 22. Incremented (++a) double local variable number 5 \rightarrow \text{KILLED}
      23. Incremented (++a) double local variable number 1 → KILLED
      24. Decremented (--a) double local variable number 5 → KILLED
      25. Decremented (--a) double local variable number 1 → KILLED

    Replaced double multiplication with division → KILLED

    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

    Replaced double operation by second member → KILLED

      10. Replaced double multiplication with division → KILLED

    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 \rightarrow SURVIVED 19. Incremented (a++) double local variable number 3 \rightarrow SURVIVED
      20. Decremented (a--) double local variable number 5 → SURVIVED
      21. Decremented (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. Decremented (--a) double local variable number 5 → KILLED
      25. Decremented (--a) double local variable number 3 → KILLED
333

    changed conditional boundary → SURVIVED
```

```
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

 Negated double local variable number 9 → KILLED

      Less or equal to less than → SURVIVED
      Less or equal to greater than → KILLED
      9. Less or equal to greater or equal \rightarrow 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. Decremented (a--) double local variable number 7 → KILLED
      15. Decremented (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. Decremented (--a) double local variable number 7 → KILLED
      19. Decremented (--a) double local variable number 9 → KILLED
      1. Substituted 2.0 with 1.0 → KILLED
      Substituted 2.0 with 1.0 → KILLED
      Replaced double division with multiplication → KILLED
      4. Replaced double division with multiplication → KILLED
      Replaced double addition with subtraction → KILLED

 Negated double local variable number 7 → KILLED

    Negated double local variable number 9 → KILLED

      Replaced double operation with first member → KILLED
      Replaced double operation with first member → KILLED

    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 \rightarrow KILLED
      14. Replaced double division with multiplication → KILLED
      15. Replaced double division with multiplication → KILLED
      16. Replaced double addition with subtraction → KILLED

    Replaced double division with modulus → KILLED

    Replaced double division with modulus → KILLED

      19. Replaced double addition with multiplication → KILLED
      20. Replaced double division with addition → KILLED
      Replaced double division with addition → KILLED
      22. Replaced double addition with division → KILLED
      23. Replaced double division with subtraction \rightarrow KILLED
334
      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 \rightarrow SURVIVED
      39. Incremented (a++) double local variable number 9 → SURVIVED 40. Decremented (a--) double local variable number 7 → SURVIVED 41. Decremented (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. Decremented (--a) double local variable number 7 → KILLED
      45. Decremented (--a) double local variable number 9 → KILLED

    Negated double local variable number 7 → KILLED

    Incremented (a++) double local variable number 7 → KILLED
    Decremented (a--) double local variable number 7 → KILLED
```

```
    Incremented (++a) double local variable number 7 → KILLED

      5. Decremented (--a) double local variable number 7 → KILLED

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

    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

    Negated double local variable number 7 → KILLED

    Negated double local variable number 9 → KILLED

      6. Incremented (a++) double local variable number 7 → SURVIVED
337
      7. Incremented (a++) double local variable number 9 → SURVIVED

    Decremented (a--) double local variable number 7 → SURVIVED
    Decremented (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. Decremented (--a) double local variable number 7 → KILLED
      13. Decremented (--a) double local variable number 9 → KILLED

    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

 Negated double local variable number 1 → KILLED

349
      7. Substituted 0 with 1 → SURVIVED
      8. Substituted 0 with -1 → SURVIVED
      Substituted 0 with 1 → SURVIVED
      10. Substituted 0 with -1 → SURVIVED
      11. Incremented (a++) double local variable number 1 \rightarrow SURVIVED
      12. Decremented (a--) double local variable number 1 \rightarrow SURVIVED
      13. Incremented (++a) double local variable number 1 → KILLED
      14. Decremented (--a) double local variable number 1 → KILLED

    removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted →

365
      SURVIVED

    negated conditional → SURVIVED

      2. removed conditional - replaced equality check with false \rightarrow SURVIVED 3. removed conditional - replaced equality check with true \rightarrow SURVIVED
      4. Negated integer local variable number 3 → SURVIVED
      5. equal to less than → SURVIVED
      equal to less or equal → SURVIVED
      7. equal to greater than → SURVIVED
366
      8. equal to greater or equal → SURVIVED
      9. equal to not equal → SURVIVED
      10. Incremented (a++) integer local variable number 3 → SURVIVED
      11. Decremented (a--) integer local variable number 3 → SURVIVED
      12. Incremented (++a) integer local variable number 3 → SURVIVED
      13. Decremented (--a) integer local variable number 3 → SURVIVED

    removed call to org/jfree/data/Range::<init> → NO_COVERAGE

      Replaced double addition with subtraction → NO COVERAGE
      removed call to org/jfree/data/Range::getLowerBound → NO_COVERAGE
      4. replaced return value with null for org/jfree/data/Range::shift →
      NO COVERAGE
      5. mutated return of Object value for org/jfree/data/Range::shift to ( if (x !=
      null) null else throw new RuntimeException ) → NO_COVERAGE

    Negated double local variable number 1 → NO_COVERAGE

    Replaced double operation with first member → NO_COVERAGE

367
      8. Replaced double operation by second member → NO_COVERAGE
      Replaced double addition with subtraction → NO_COVERAGE
      10. Replaced double addition with multiplication \rightarrow NO_COVERAGE

    Replaced double addition with division → NO_COVERAGE

      12. Replaced double addition with modulus → NO_COVERAGE
      13. Incremented (a++) double local variable number 1 \rightarrow NO_{COVERAGE}
      14. Decremented (a--) double local variable number 1 \rightarrow NO_{COVERAGE}
      15. Incremented (++a) double local variable number 1 \rightarrow NO COVERAGE
      16. Decremented (--a) double local variable number 1 \rightarrow NO_{COVERAGE}
368

    Replaced double addition with subtraction → NO_COVERAGE
```

```
removed call to org/jfree/data/Range::getUpperBound → NO_COVERAGE

    Negated double local variable number 1 → NO_COVERAGE

    Replaced double operation with first member → NO_COVERAGE

    Replaced double operation by second member → NO_COVERAGE

      Replaced double addition with subtraction → NO_COVERAGE
      7. Replaced double addition with multiplication → NO_COVERAGE
      Replaced double addition with division → NO_COVERAGE
      9. Replaced double addition with modulus \rightarrow NO_\overline{C}OVERAGE
      10. Incremented (a++) double local variable number 1 → NO_COVERAGE
      11. Decremented (a--) double local variable number 1 → NO_COVERAGE
      12. Incremented (++a) double local variable number 1 → NO_COVERAGE
      13. Decremented (--a) double local variable number 1 \rightarrow NO_{COVERAGE}

    replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument

      → KILLED
      removed call to org/jfree/data/Range::<init> → KILLED
      3. removed call to org/jfree/data/Range::getLowerBound → KILLED
4. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → SURVIVED
      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

    replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument

      → KILLED
      removed call to org/jfree/data/Range::getUpperBound → KILLED
      3. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
      4. Negated double local variable number 1 → KILLED
372
      5. Incremented (a++) double local variable number 1 → KILLED
      6. Decremented (a--) double local variable number 1 → KILLED
      7. Incremented (++a) double local variable number 1 → KILLED
      8. Decremented (--a) double local variable number 1 → KILLED
      1. Negated double local variable number 1 → KILLED
      2. Incremented (a++) double local variable number 1 \rightarrow SURVIVED
      3. Decremented (a--) double local variable number 1 → SURVIVED
373
      4. Incremented (++a) double local variable number 1 → KILLED

    Decremented (--a) double local variable number 1 → KILLED

    changed conditional boundary → SURVIVED

      Substituted 0.0 with 1.0 → SURVIVED
      3. negated conditional → SURVIVED
      4. removed conditional - replaced comparison check with false → SURVIVED
      5. removed conditional - replaced comparison check with true → SURVIVED
      6. Negated double local variable number 0 → SURVIVED
      7. Substituted 0.0 with 1.0 → SURVIVED
      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 → SURVIVED
      13. Less or equal to greater or equal → SURVIVED
      14. Less or equal to equal → SURVIVED
      15. Less or equal to not equal → SURVIVED
      16. Incremented (a++) double local variable number 0 \to \text{KILLED} 17. Decremented (a--) double local variable number 0 \to \text{KILLED} 18. Incremented (++a) double local variable number 0 \to \text{KILLED}
      19. Decremented (--a) double local variable number 0 → KILLED
<u>388</u>

    replaced call to java/lang/Math::max with argument → KILLED

      2. Substituted 0.0 with 1.0 → SURVIVED

    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

    Negated double local variable number 0 → KILLED

      8. Negated double local variable number 2 → KILLED

    Replaced double operation with first member → KILLED

    Replaced double operation by second member → KILLED
```

```
    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 → SURVIVED
      16. Substituted 0.0 with -1.0 → SURVIVED
      17. Substituted 0.0 with 1.0 → SURVIVED
      18. Substituted 0.0 with -1.0 → SURVIVED
      19. Incremented (a++) double local variable number 0 → SURVIVED
      20. Incremented (a++) double local variable number 2 → SURVIVED
      21. Decremented (a--) double local variable number 0 → SURVIVED
      22. Decremented (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. Decremented (--a) double local variable number 0 → KILLED
      26. Decremented (--a) double local variable number 2 → KILLED

    changed conditional boundary → SURVIVED

    Substituted 0.0 with 1.0 → SURVIVED

      negated conditional → SURVIVED

    removed conditional - replaced comparison check with false → SURVIVED

      removed conditional - replaced comparison check with true → SURVIVED
      Negated double local variable number 0 → SURVIVED
      7. Substituted 0.0 with 1.0 → SURVIVED
      Substituted 0.0 with -1.0 → SURVIVED
      9. Substituted 0.0 with 1.0 → SURVIVED
390
      10. Substituted 0.0 with -1.0 → SURVIVED

    greater or equal to less than → SURVIVED

      12. greater or equal to less or equal → SURVIVED
      13. greater or equal to greater than → SURVIVED
      14. greater or equal to equal → SURVIVED
      15. greater or equal to not equal → SURVIVED
      16. Incremented (a++) double local variable number 0 → SURVIVED
      17. Decremented (a--) double local variable number 0 → KILLED
      18. Incremented (++a) double local variable number 0 → KILLED
      19. Decremented (--a) double local variable number 0 → KILLED

    replaced call to java/lang/Math::min with argument → SURVIVED

      Substituted 0.0 with 1.0 → SURVIVED

    Replaced double addition with subtraction → KILLED

      4. removed call to java/lang/Math::min → SURVIVED
      5. replaced double return with 0.0d for
      org/jfree/data/Range::shiftWithNoZeroCrossing \rightarrow SURVIVED
      6. replaced return of double value with -(x + 1) for
      org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
      7. Negated double local variable number 0 → SURVIVED

 Negated double local variable number 2 → KILLED

      Replaced double operation with first member → KILLED

    Replaced double operation by second member → SURVIVED

      11. Replaced double addition with subtraction \rightarrow KILLED
      12. Replaced double addition with multiplication → KILLED
391
      13. Replaced double addition with division → KILLED
      14. Replaced double addition with modulus → SURVIVED
      15. Substituted 0.0 with 1.0 → SURVIVED
      16. Substituted 0.0 with -1.0 → KILLED
      17. Substituted 0.0 with 1.0 → SURVIVED
      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. Decremented (a--) double local variable number 0 → SURVIVED
      22. Decremented (a--) double local variable number 2 → SURVIVED
      23. Incremented (++a) double local variable number 0 \rightarrow SURVIVED
      24. Incremented (++a) double local variable number 2 → SURVIVED
      25. Decremented (--a) double local variable number 0 → KILLED
      26. Decremented (--a) double local variable number 2 → KILLED
394

    Replaced double addition with subtraction → NO COVERAGE

      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

    Negated double local variable number 0 → NO_COVERAGE

      Negated double local variable number 2 → NO_COVERAGE
      Replaced double operation with first member → NO_COVERAGE

    Replaced double operation by second member → NO_COVERAGE

      Replaced double addition with subtraction → NO_COVERAGE
      9. Replaced double addition with multiplication → NO_COVERAGE
      10. Replaced double addition with division \rightarrow NO_COVERAGE

    Replaced double addition with modulus → NO_COVERAGE

      12. Incremented (a++) double local variable number 0 \rightarrow NO_{COVERAGE}
      13. Incremented (a++) double local variable number 2 \rightarrow NO COVERAGE
      14. Decremented (a--) double local variable number \emptyset \rightarrow NO COVERAGE

    Decremented (a--) double local variable number 2 → NO_COVERAGE

    Incremented (++a) double local variable number 0 → NO_COVERAGE

      17. Incremented (++a) double local variable number 2 → NO_COVERAGE
      18. Decremented (--a) double local variable number 0 → NO_COVERAGE

    Decremented (--a) double local variable number 2 → NO_COVERAGE

    removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted →

409
      SURVIVED

    changed conditional boundary → SURVIVED

      2. Substituted 0.0 with 1.0 → SURVIVED
      negated conditional → KILLED

    removed conditional - replaced comparison check with false → SURVIVED

      5. removed conditional - replaced comparison check with true \rightarrow KILLED
      6. Negated double local variable number 1 → KILLED
      7. Substituted 0.0 with 1.0 → SURVIVED
      Substituted 0.0 with -1.0 → SURVIVED
      9. Substituted 0.0 with 1.0 → SURVIVED
410
      10. Substituted 0.0 with -1.0 → SURVIVED

 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 → KILLED
      15. greater or equal to not equal → SURVIVED
      16. Incremented (a++) double local variable number 1 \rightarrow KILLED 17. Decremented (a--) double local variable number 1 \rightarrow KILLED
      18. Incremented (++a) double local variable number 1 → KILLED
      19. Decremented (--a) double local variable number 1 → KILLED
411

    removed call to java/lang/IllegalArgumentException::<init> → KILLED

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

    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 \rightarrow KILLED
      Replaced double operation by second member → SURVIVED
      9. Replaced double multiplication with division → KILLED
      10. Replaced double multiplication with modulus 
ightarrow KILLED

    Replaced double multiplication with addition → KILLED

      12. Replaced double multiplication with subtraction → KILLED
      13. Incremented (a++) double local variable number 1 → KILLED
      14. Decremented (a--) double local variable number 1 → KILLED
      15. Incremented (++a) double local variable number 1 \rightarrow KILLED
      16. Decremented (--a) double local variable number 1 \rightarrow KILLED
414

    Replaced double multiplication with division → KILLED

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

 Negated double local variable number 1 → KILLED

    Replaced double operation with first member → KILLED

      5. Replaced double operation by second member → KILLED
      Replaced double multiplication with division → KILLED
      7. Replaced double multiplication with modulus → KILLED
      Replaced double multiplication with addition → KILLED
```

```
Replaced double multiplication with subtraction → KILLED
      10. Incremented (a++) double local variable number 1 \rightarrow SURVIVED
      11. Decremented (a--) double local variable number 1 \rightarrow SURVIVED
      12. Incremented (++a) double local variable number 1 → KILLED
      13. Decremented (--a) double local variable number 1 \rightarrow KILLED

    negated conditional → KILLED

      removed conditional - replaced equality check with false → KILLED

    removed conditional - replaced equality check with true → KILLED

      4. not equal to less than → KILLED
426
      5. not equal to less or equal \rightarrow KILLED
      6. not equal to greater than → SURVIVED
      7. not equal to greater or equal \rightarrow KILLED
      8. not equal to equal → KILLED

    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) \rightarrow KILLED
427
      4. Substituted 0 with 1 → KILLED
      5. Substituted 0 with -1 → KILLED
      6. Substituted 0 with 1 → KILLED

 Substituted 0 with -1 → KILLED

    negated conditional → KILLED

      removed conditional - replaced equality check with false → KILLED
      3. removed conditional - replaced equality check with true 
ightarrow KILLED
      4. Negated double field lower → KILLED
      Negated double field lower → KILLED
      equal to less than → KILLED

 equal to less or equal → KILLED

      equal to greater than → KILLED
      9. equal to greater or equal \rightarrow SURVIVED
430
      10. equal to not equal → KILLED
      11. Incremented (a++) double field lower → SURVIVED
      12. Incremented (a++) double field lower → SURVIVED
      13. Decremented (a--) double field lower → SURVIVED
      14. Decremented (a--) double field lower → SURVIVED
      15. Incremented (++a) double field lower → KILLED
      16. Incremented (++a) double field lower → KILLED
      17. Decremented (--a) double fieldlower → KILLED
      18. Decremented (--a) double fieldlower → KILLED

    replaced boolean return with true for org/jfree/data/Range::equals → KILLED

      Substituted 0 with 1 → KILLED
      3. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow KILLED
431
      4. Substituted 0 with 1 → KILLED
      Substituted 0 with -1 → KILLED
      6. Substituted 0 with 1 → KILLED
      7. Substituted 0 with -1 \rightarrow KILLED

    negated conditional → KILLED

      removed conditional - replaced equality check with false → KILLED

    removed conditional - replaced equality check with true → KILLED

 Negated double field upper → KILLED

      Negated double field upper → KILLED
      6. equal to less than → KILLED
      7. equal to less or equal → KILLED
      8. equal to greater than → KILLED
      9. equal to greater or equal → SURVIVED
433
      10. equal to not equal → KILLED
      11. Incremented (a++) double field upper → SURVIVED 12. Incremented (a++) double field upper → SURVIVED 13. Decremented (a--) double field upper → SURVIVED
      14. Decremented (a--) double field upper → SURVIVED
      15. Incremented (++a) double field upper → KILLED
      16. Incremented (++a) double field upper → KILLED
      17. Decremented (--a) double fieldupper → KILLED
      18. Decremented (--a) double fieldupper → KILLED
434

    replaced boolean return with true for org/jfree/data/Range::equals → KILLED

      Substituted 0 with 1 → KILLED
```

```
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

 Substituted 0 with -1 → KILLED

    replaced boolean return with false for org/jfree/data/Range::equals → KILLED

      Substituted 1 with 0 → KILLED
      3. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow KILLED
      4. Substituted 1 with 0 → KILLED
436
      5. Substituted 1 with -1 → SURVIVED
      6. Substituted 1 with -1 → SURVIVED

 Substituted 1 with 2 → KILLED

      8. Substituted 1 with 0 → KILLED

    replaced boolean return with false for org/jfree/data/Range::isNaNRange →

      NO COVERAGE
      replaced boolean return with true for org/jfree/data/Range::isNaNRange →
      KILLED
      Substituted 1 with 0 → NO_COVERAGE
      Substituted 0 with 1 → KILLED
      negated conditional → SURVIVED
      negated conditional → NO_COVERAGE
      7. removed call to java/lang/Double::isNaN → SURVIVED
8. removed call to java/lang/Double::isNaN → NO_COVERAGE
      9. removed conditional - replaced equality check with false \rightarrow SURVIVED
      10. removed conditional - replaced equality check with false → NO_COVERAGE
      11. removed conditional - replaced equality check with true 
ightarrow SUR\overline{	extsf{VIVED}}
      12. removed conditional - replaced equality check with true 
ightarrow NO_COVERAGE
      13. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow NO_COVERAGE
      14. replaced return of integer sized value with (x == 0 ? 1 : 0) \rightarrow KI\overline{L}LED
      15. Negated double field lower → SURVIVED
      16. Negated double field upper → NO_COVERAGE
      17. Substituted 0 with 1 → KILLED
      18. Substituted 1 with 0 → NO_COVERAGE
      19. Substituted 1 with -1 → N\overline{O}_COVERAGE
      20. Substituted 0 with -1 → KILLED
      21. Substituted 1 with -1 → NO_COVERAGE
448
      22. Substituted 1 with 2 → NO_COVERAGE
      23. Substituted 0 with 1 → KILLED
      24. Substituted 1 with 0 → NO_COVERAGE
      25. Substituted 0 with -1 → \overline{KILLED}
      26. equal to less than → SURVIVED
      27. equal to less than → NO COVERAGE
      28. equal to less or equal → SURVIVED
      29. equal to less or equal → NO_COVERAGE
      30. equal to greater than → SURVIVED
      31. equal to greater than → NO_COVERAGE
      32. equal to greater or equal → SURVIVED
      33. equal to greater or equal → NO_COVERAGE
      34. equal to not equal → SURVIVED
      35. equal to not equal → NO_COVERAGE
36. Incremented (a++) double field lower → SURVIVED
37. Incremented (a++) double field upper → NO_COVERAGE
      38. Decremented (a--) double field lower → SURVIVED
      39. Decremented (a--) double field upper → NO_COVERAGE
      40. Incremented (++a) double field lower → SURVIVED
      41. Incremented (++a) double field upper → NO_COVERAGE
      42. Decremented (--a) double fieldlower → SURVIVED
      43. Decremented (--a) double fieldupper → NO COVERAGE

    removed call to java/lang/Double::doubleToLongBits → KILLED
    Negated double field lower → KILLED

      3. Incremented (a++) double field lower \rightarrow SURVIVED
460
      4. Decremented (a--) double field lower → SURVIVED
      5. Incremented (++a) double field lower → KILLED
      6. Decremented (--a) double fieldlower → KILLED
461
      1. Substituted 32 with 33 → KILLED
      2. Replaced Unsigned Shift Right with Shift Left → KILLED
```

```
Replaced XOR with AND → KILLED

    Negated long local variable number 2 → SURVIVED

      5. Negated long local variable number 2 \rightarrow KILLED 6. Substituted 32 with 1 \rightarrow KILLED

 Substituted 32 with 0 → KILLED

      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 \rightarrow SURVIVED
      13. Incremented (a++) long local variable number 1 → SURVIVED
      14. Decremented (a--) long local variable number 2 → KILLED
      15. Decremented (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. Decremented (--a) long local variable number 1 \rightarrow KILLED
      19. Decremented (--a) long local variable number 1 → KILLED

    removed call to java/lang/Double::doubleToLongBits → KILLED

      Negated double field upper → KILLED
      3. Incremented (a++) double field upper → SURVIVED
462

    Decremented (a--) double field upper → SURVIVED

    Incremented (++a) double field upper → KILLED

      6. Decremented (--a) double fieldupper → KILLED
463
      1. Substituted 29 with 30 → KILLED
      2. Substituted 32 with 33 → KILLED

    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 1 → KILLED
      8. Negated long local variable number 2 → SURVIVED
      9. Negated long local variable number 2 → KILLED
      10. Replaced integer operation with first member → KILLED
      11. Replaced integer operation with first member \rightarrow KILLED
      12. Replaced integer operation by second member \rightarrow KILLED

    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
      22. Substituted 29 with 1 → KILLED
      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. Decremented (a--) integer local variable number 1 → SURVIVED 38. Decremented (a--) long local variable number 2 → KILLED
      39. Decremented (a--) long local variable number 2 \rightarrow 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. Decremented (--a) integer local variable number 3 → KILLED
```

```
44. Decremented (--a) long local variable number 1 → KILLED
       45. Decremented (--a) long local variable number 1 → KILLED

    replaced int return with 0 for org/jfree/data/Range::hashCode → KILLED

       2. replaced return of integer sized value with (x == 0?1:0) \rightarrow KILLED
       3. Negated integer local variable number 1 → KILLED
464

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

       5. Decremented (a--) integer local variable number 1 → SURVIVED 6. Incremented (++a) integer local variable number 3 → KILLED 7. Decremented (--a) integer local variable number 3 → KILLED

    removed call to java/lang/StringBuilder::<init> → KILLED
    replaced return value with "" for org/jfree/data/Range::toString → KILLED

       3. removed call to java/lang/StringBuilder::append → KILLED

 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 ) \rightarrow KILLED
       9. replaced call to java/lang/StringBuilder::append with receiver \rightarrow 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
475
       14. Negated double field upper → KILLED
       15. Incremented (a++) double field lower \rightarrow SURVIVED
       16. Incremented (a++) double field upper → SURVIVED

    Decremented (a--) double field lower → SURVIVED

 Decremented (a--) double field upper → SURVIVED

       19. Incremented (++a) double field lower → KILLED
       20. Incremented (++a) double field upper → KILLED
       21. Decremented (--a) double fieldlower → KILLED
       22. Decremented (--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

- ARGŪMENT 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 VALUESEXPERIMENTAL 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

- 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
- REMOVE_INCREMENTS_MUTATOR
- RETURN VALS MUTATOR
- ROR_1_MUTATOR
 ROR_2_MUTATOR
 ROR_3_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.RangeTestUnmutated (40 ms)

Report generated by PIT 1.4.11