Range.java

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

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

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

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

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

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

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

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

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

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

Mutations

1. changed conditional boundary → KILLED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED
5. Negated double local variable number 1 → KILLED
6. Negated double local variable number 3 → KILLED
7. Less or equal to less than → KILLED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED

90
10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Incremented (a++) double local variable number 3 → KILLED
14. 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

1. removed call to java/lang/StringBuilder::<init> → SURVIVED
2. removed call to java/lang/StringBuilder::append → SURVIVED
3. removed call to java/lang/StringBuilder::toString → SURVIVED
4. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED

91
5. Negated double local variable number 1 → SURVIVED
6. Incremented (a++) double local variable number 1 → SURVIVED
7. 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

1. removed call to java/lang/StringBuilder::append → SURVIVED
2. removed call to java/lang/StringBuilder::append → SURVIVED
3. removed call to java/lang/StringBuilder::append → SURVIVED
4. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
5. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED

92
6. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
7. Negated double local variable number 3 → SURVIVED
8. Incremented (a++) double local variable number 3 → SURVIVED
9. 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
1. removed call to java/lang/IllegalArgumentException::<init> → SURVIVED

1. Removed assignment to member variable lower → KILLED
2. Negated double local variable number 1 → KILLED

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

1. Removed assignment to member variable upper → KILLED
2. Negated double local variable number 3 → KILLED

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

105
1. replaced double return with 0.0d for org/jfree/data/Range::getLowerBound → KILLED

2. replaced return of double value with -(x + 1) for org/jfree/data/Range::getLowerBound → KILLED
3. Negated double field lower → KILLED
4. Incremented (a++) double field lower → KILLED
5. Decremented (a--) double field lower → KILLED
6. Incremented (++a) double field lower → KILLED
7. Decremented (--a) double fieldlower → KILLED

1. replaced double return with 0.0d for org/jfree/data/Range::getUpperBound → KILLED
2. replaced return of double value with -(x + 1) for org/jfree/data/Range::getUpperBound → KILLED

114 3. Negated double field upper → KILLED
4. Incremented (a++) double field upper → KILLED
5. Decremented (a--) double field upper → KILLED
6. Incremented (++a) double field upper → KILLED
7. Decremented (--a) double fieldupper → KILLED

1. Replaced double subtraction with addition → KILLED
2. replaced double return with 0.0d for org/jfree/data/Range::getLength → KILLED
3. replaced return of double value with -(x + 1) for org/jfree/data/Range::getLength → KILLED
4. Negated double field upper → KILLED
5. Negated double field lower → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation by second member → KILLED
8. Replaced double subtraction with addition → KILLED

123 9. Replaced double subtraction with multiplication → KILLED
10. Replaced double subtraction with division → KILLED
11. Replaced double subtraction with modulus → KILLED
12. Incremented (a++) double field upper → KILLED
13. Incremented (a++) double field lower → KILLED
14. Decremented (a--) double field upper → KILLED
15. Decremented (a--) double field lower → 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

132 1. Substituted 2.0 with 1.0 → SURVIVED
2. Substituted 2.0 with 1.0 → SURVIVED
3. Replaced double division with multiplication → SURVIVED
4. Replaced double division with multiplication → SURVIVED
5. Replaced double addition with subtraction → SURVIVED
6. replaced double return with 0.0d for org/jfree/data/Range::getCentralValue → SURVIVED
7. replaced return of double value with -(x + 1) for org/jfree/data/Range::getCentralValue → SURVIVED
8. Negated double field lower → SURVIVED
9. Negated double field upper → SURVIVED
10. Replaced double operation with first member → SURVIVED
11. Replaced double operation with first member → SURVIVED
12. Replaced double operation with first member → SURVIVED
13. Replaced double operation by second member → SURVIVED
14. Replaced double operation by second member → SURVIVED
15. Replaced double operation by second member → SURVIVED
16. Replaced double division with multiplication → SURVIVED
17. Replaced double division with multiplication → SURVIVED
18. Replaced double addition with subtraction → SURVIVED
19. Replaced double division with modulus → SURVIVED
20. Replaced double division with modulus → SURVIVED
21. Replaced double addition with multiplication → SURVIVED
22. Replaced double division with addition → SURVIVED
23. Replaced double division with addition → SURVIVED
24. Replaced double addition with division → SURVIVED
25. Replaced double division with subtraction → SURVIVED
26. Replaced double division with subtraction → SURVIVED
27. Replaced double addition with modulus → SURVIVED

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

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

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

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED
5. Negated double local variable number 1 → KILLED
6. Negated double field lower → SURVIVED
7. greater than to less than → KILLED
8. greater than to less or equal → KILLED
9. greater than to greater or equal → SURVIVED

157

10. greater than to equal → KILLED
11. greater than to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Incremented (a++) double field lower → KILLED
14. Decremented (a--) double local variable number 1 → KILLED
15. Decremented (a--) double field lower → KILLED
16. Incremented (++a) double local variable number 1 → KILLED
17. Incremented (++a) double field lower → KILLED
18. Decremented (--a) double local variable number 1 → KILLED
19. Decremented (--a) double fieldlower → KILLED

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED
2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED
3. changed conditional boundary → KILLED
4. Substituted 1 with 0 → KILLED
5. Substituted 0 with 1 → KILLED
6. negated conditional → KILLED
7. removed conditional - replaced comparison check with false → KILLED
8. removed conditional - replaced comparison check with true → KILLED
9. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
10. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
11. Negated double local variable number 3 → KILLED
12. Negated double field lower → SURVIVED
13. Substituted 0 with 1 → KILLED
14. Substituted 1 with 0 → KILLED
15. Substituted 1 with -1 → SURVIVED

158

16. Substituted 0 with -1 → KILLED
17. Substituted 1 with -1 → SURVIVED
18. Substituted 1 with 2 → KILLED
19. Substituted 0 with 1 → KILLED
20. Substituted 1 with 0 → KILLED
21. Substituted 0 with -1 → KILLED
22. Less or equal to less than → KILLED
23. Less or equal to greater than → KILLED
24. Less or equal to greater or equal → KILLED
25. Less or equal to equal → KILLED
26. Less or equal to not equal → KILLED
27. Incremented (a++) double local variable number 3 → SURVIVED
28. Incremented (a++) double field lower → KILLED
29. Decremented (a--) double local variable number 3 → SURVIVED
30. Decremented (a--) double field lower → KILLED
31. Incremented (++a) double local variable number 3 → KILLED
32. Incremented (++a) double field lower → KILLED
33. Decremented (--a) double local variable number 3 → KILLED
34. Decremented (--a) double fieldlower → KILLED

161

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

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

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

176

3. removed call to org/jfree/data/Range::getLowerBound → NO_COVERAGE
4. removed call to org/jfree/data/Range::getUpperBound → NO_COVERAGE
5. removed call to org/jfree/data/Range::intersects → NO_COVERAGE
6. replaced return of integer sized value with (x == 0 ? 1 : 0) → NO_COVERAGE

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

188

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

189 1. negated conditional → KILLED
2. removed call to org/jfree/data/Range::contains → SURVIVED
3. removed conditional - replaced equality check with false → KILLED
4. removed conditional - replaced equality check with true → SURVIVED
5. Negated double local variable number 1 → KILLED
6. not equal to less than → SURVIVED
7. not equal to less or equal → KILLED
8. not equal to greater than → SURVIVED
9. not equal to greater or equal → KILLED
10. not equal to equal → KILLED
11. Incremented (a++) double local variable number 1 → KILLED
12. Decremented (a--) double local variable number 1 → KILLED
13. Incremented (++a) double local variable number 1 → KILLED
14. Decremented (--a) double local variable number 1 → KILLED

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED
5. Negated double local variable number 1 → KILLED
6. Negated double field upper → KILLED
7. Less or equal to less than → SURVIVED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
190 10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Incremented (a++) double field upper → KILLED
14. Decremented (a--) double local variable number 1 → SURVIVED
15. Decremented (a--) double field upper → KILLED
16. Incremented (++a) double local variable number 1 → KILLED
17. Incremented (++a) double field upper → KILLED
18. Decremented (--a) double local variable number 1 → KILLED
19. Decremented (--a) double fieldupper → KILLED

1. Negated double field upper → KILLED
2. Incremented (a++) double field upper → KILLED
191 3. Decremented (a--) double field upper → KILLED
4. Incremented (++a) double field upper → KILLED
5. Decremented (--a) double fieldupper → KILLED

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED
5. Negated double local variable number 1 → KILLED
6. Negated double field lower → KILLED
7. greater or equal to less than → KILLED
8. greater or equal to less or equal → KILLED
9. greater or equal to greater than → SURVIVED
193 10. greater or equal to equal → KILLED
11. greater or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → SURVIVED
13. Incremented (a++) double field lower → KILLED
14. Decremented (a--) double local variable number 1 → SURVIVED
15. Decremented (a--) double field lower → KILLED
16. Incremented (++a) double local variable number 1 → KILLED
17. Incremented (++a) double field lower → KILLED
18. Decremented (--a) double local variable number 1 → SURVIVED
19. Decremented (--a) double fieldlower → KILLED

1. Negated double field lower → KILLED
2. Incremented (a++) double field lower → KILLED
194 3. Decremented (a--) double field lower → KILLED
4. Incremented (++a) double field lower → KILLED
5. Decremented (--a) double fieldlower → KILLED

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

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

	2. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to (if (x != null) null else throw new RuntimeException) → KILLED
247	1. negated conditional → KILLED 2. removed conditional - replaced equality check with false → KILLED 3. removed conditional - replaced equality check with true → KILLED 4. not equal to equal → KILLED
248	1. negated conditional → KILLED 2. removed call to org/jfree/data/Range::isNaNRange → SURVIVED 3. removed conditional - replaced equality check with false → SURVIVED 4. removed conditional - replaced equality check with true → KILLED 5. equal to less than → KILLED 6. equal to less or equal → SURVIVED 7. equal to greater than → KILLED 8. equal to greater or equal → SURVIVED 9. equal to not equal → KILLED
249	1. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to (if (x != null) null else throw new RuntimeException) → NO_COVERAGE
251	1. replaced return value with null for org/jfree/data/Range::combineIgnoringNaN → KILLED 2. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to (if (x != null) null else throw new RuntimeException) → KILLED
253	1. replaced call to org/jfree/data/Range::min with argument → KILLED 2. removed call to org/jfree/data/Range::getLowerBound → KILLED 3. removed call to org/jfree/data/Range::getLowerBound → KILLED 4. removed call to org/jfree/data/Range::min → KILLED
254	1. replaced call to org/jfree/data/Range::max with argument → KILLED 2. removed call to org/jfree/data/Range::getUpperBound → KILLED 3. removed call to org/jfree/data/Range::getUpperBound → KILLED 4. removed call to org/jfree/data/Range::max → KILLED
255	1. negated conditional → SURVIVED 2. negated conditional → NO_COVERAGE 3. removed call to java/lang/Double::isNaN → SURVIVED 4. removed call to java/lang/Double::isNaN → NO_COVERAGE 5. removed conditional - replaced equality check with false → SURVIVED 6. removed conditional - replaced equality check with false → NO_COVERAGE 7. removed conditional - replaced equality check with true → SURVIVED 8. removed conditional - replaced equality check with true → NO_COVERAGE 9. Negated double local variable number 2 → SURVIVED 10. Negated double local variable number 4 → NO_COVERAGE 11. equal to less than → SURVIVED 12. equal to less than → NO_COVERAGE 13. equal to less or equal → SURVIVED 14. equal to less or equal → NO_COVERAGE 15. equal to greater than → SURVIVED 16. equal to greater than → NO_COVERAGE 17. equal to greater or equal → SURVIVED 18. equal to greater or equal → NO_COVERAGE 19. equal to not equal → SURVIVED 20. equal to not equal → NO_COVERAGE 21. Incremented (a++) double local variable number 2 → KILLED 22. Incremented (a++) double local variable number 4 → NO_COVERAGE 23. Decremented (a--) double local variable number 2 → KILLED 24. Decremented (a--) double local variable number 4 → NO_COVERAGE 25. Incremented (++a) double local variable number 2 → KILLED 26. Incremented (++a) double local variable number 4 → NO_COVERAGE 27. Decremented (--a) double local variable number 2 → KILLED 28. Decremented (--a) double local variable number 4 → NO_COVERAGE
256	1. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to (if (x != null) null else throw new RuntimeException) → NO_COVERAGE
258	1. removed call to org/jfree/data/Range::<init> → KILLED 2. replaced return value with null for org/jfree/data/Range::combineIgnoringNaN → KILLED 3. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to (if (x != null) null else throw new RuntimeException) → KILLED

4. Negated double local variable number 2 → KILLED
5. Negated double local variable number 4 → KILLED
6. Incremented (a++) double local variable number 2 → SURVIVED
7. Incremented (a++) double local variable number 4 → SURVIVED
8. 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

271

1. negated conditional → KILLED
2. removed call to java/lang/Double::isNaN → SURVIVED
3. removed conditional - replaced equality check with false → SURVIVED
4. removed conditional - replaced equality check with true → KILLED
5. Negated double local variable number 0 → SURVIVED
6. equal to less than → KILLED
7. equal to less or equal → SURVIVED
8. equal to greater than → KILLED
9. equal to greater or equal → SURVIVED
10. equal to not equal → KILLED
11. Incremented (a++) double local variable number 0 → KILLED
12. 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

272

1. replaced double return with 0.0d for org/jfree/data/Range::min → NO_COVERAGE
2. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → NO_COVERAGE
3. Negated double local variable number 2 → NO_COVERAGE
4. Incremented (a++) double local variable number 2 → NO_COVERAGE
5. Decremented (a--) double local variable number 2 → NO_COVERAGE
6. Incremented (++a) double local variable number 2 → NO_COVERAGE
7. Decremented (--a) double local variable number 2 → NO_COVERAGE

274

1. negated conditional → KILLED
2. removed call to java/lang/Double::isNaN → KILLED
3. removed conditional - replaced equality check with false → KILLED
4. removed conditional - replaced equality check with true → KILLED
5. Negated double local variable number 2 → SURVIVED
6. equal to less than → KILLED
7. equal to less or equal → SURVIVED
8. equal to greater than → KILLED
9. equal to greater or equal → KILLED
10. equal to not equal → KILLED
11. Incremented (a++) double local variable number 2 → KILLED
12. 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

275

1. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED
2. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → KILLED
3. Negated double local variable number 0 → KILLED
4. Incremented (a++) double local variable number 0 → SURVIVED
5. Decremented (a--) double local variable number 0 → SURVIVED
6. Incremented (++a) double local variable number 0 → KILLED
7. Decremented (--a) double local variable number 0 → KILLED

277

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

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

1. removed call to org/jfree/data/Range::<init> → NO_COVERAGE
2. replaced return value with null for org/jfree/data/Range::expandToInclude → NO_COVERAGE
3. mutated return of Object value for org/jfree/data/Range::expandToInclude to (if (x != null) null else throw new RuntimeException) → NO_COVERAGE
4. Negated double local variable number 1 → NO_COVERAGE
5. Negated double local variable number 1 → NO_COVERAGE
303 6. Incremented (a++) double local variable number 1 → NO_COVERAGE
7. Incremented (a++) double local variable number 1 → NO_COVERAGE
8. Decremented (a--) double local variable number 1 → NO_COVERAGE
9. Decremented (a--) double local variable number 1 → NO_COVERAGE
10. Incremented (++a) double local variable number 1 → NO_COVERAGE
11. Incremented (++a) double local variable number 1 → NO_COVERAGE
12. Decremented (--a) double local variable number 1 → NO_COVERAGE
13. Decremented (--a) double local variable number 1 → NO_COVERAGE

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → KILLED
4. removed conditional - replaced comparison check with false → KILLED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → SURVIVED
7. greater or equal to less than → KILLED
305 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

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

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed call to org/jfree/data/Range::getUpperBound → SURVIVED
4. removed conditional - replaced comparison check with false → SURVIVED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → SURVIVED
7. Less or equal to less than → SURVIVED
308 8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
10. Less or equal to equal → KILLED
11. Less or equal to not equal → SURVIVED
12. Incremented (a++) double local variable number 1 → SURVIVED
13. Decremented (a--) double local variable number 1 → SURVIVED
14. Incremented (++a) double local variable number 1 → SURVIVED
15. Decremented (--a) double local variable number 1 → SURVIVED

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

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

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

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

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

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

1. Replaced double multiplication with division → KILLED
2. Replaced double subtraction with addition → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
4. Negated double local variable number 5 → KILLED
5. Negated double local variable number 1 → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → KILLED
9. Replaced double operation by second member → KILLED
10. Replaced double multiplication with division → KILLED
11. Replaced double subtraction with addition → KILLED
12. Replaced double multiplication with modulus → KILLED

331 13. Replaced double subtraction with multiplication → KILLED
14. Replaced double multiplication with addition → KILLED
15. Replaced double subtraction with division → KILLED
16. Replaced double multiplication with subtraction → KILLED
17. Replaced double subtraction with modulus → KILLED
18. Incremented (a++) double local variable number 5 → KILLED
19. Incremented (a++) double local variable number 1 → SURVIVED
20. Decremented (a--) double local variable number 5 → KILLED
21. Decremented (a--) double local variable number 1 → SURVIVED
22. Incremented (++a) double local variable number 5 → KILLED
23. Incremented (++a) double local variable number 1 → KILLED
24. Decremented (--a) double local variable number 5 → KILLED
25. Decremented (--a) double local variable number 1 → KILLED

1. Replaced double multiplication with division → KILLED
2. Replaced double addition with subtraction → KILLED
3. removed call to org/jfree/data/Range::getUpperBound → KILLED
4. Negated double local variable number 5 → KILLED
5. Negated double local variable number 3 → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → KILLED
9. Replaced double operation by second member → KILLED
10. Replaced double multiplication with division → KILLED
11. Replaced double addition with subtraction → KILLED
12. Replaced double multiplication with modulus → KILLED

332 13. Replaced double addition with multiplication → KILLED
14. Replaced double multiplication with addition → KILLED
15. Replaced double addition with division → KILLED
16. Replaced double multiplication with subtraction → KILLED
17. Replaced double addition with modulus → KILLED
18. Incremented (a++) double local variable number 5 → SURVIVED
19. Incremented (a++) double local variable number 3 → SURVIVED
20. 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 1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED

5. Negated double local variable number 7 → KILLED
6. Negated double local variable number 9 → KILLED
7. Less or equal to less than → SURVIVED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 7 → KILLED
13. Incremented (a++) double local variable number 9 → KILLED
14. 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
2. Substituted 2.0 with 1.0 → KILLED
3. Replaced double division with multiplication → KILLED
4. Replaced double division with multiplication → KILLED
5. Replaced double addition with subtraction → KILLED
6. Negated double local variable number 7 → KILLED
7. Negated double local variable number 9 → KILLED
8. Replaced double operation with first member → KILLED
9. Replaced double operation with first member → KILLED
10. Replaced double operation with first member → KILLED
11. Replaced double operation by second member → KILLED
12. Replaced double operation by second member → KILLED
13. Replaced double operation by second member → KILLED
14. Replaced double division with multiplication → KILLED
15. Replaced double division with multiplication → KILLED
16. Replaced double addition with subtraction → KILLED
17. Replaced double division with modulus → KILLED
18. Replaced double division with modulus → KILLED
19. Replaced double addition with multiplication → KILLED
20. Replaced double division with addition → KILLED
21. Replaced double division with addition → KILLED
22. Replaced double addition with division → KILLED

334 23. Replaced double division with subtraction → KILLED
24. Replaced double division with subtraction → KILLED
25. Replaced double addition with modulus → KILLED
26. Substituted 2.0 with 1.0 → KILLED
27. Substituted 2.0 with 1.0 → KILLED
28. Substituted 2.0 with 0.0 → KILLED
29. Substituted 2.0 with 0.0 → KILLED
30. Substituted 2.0 with -1.0 → KILLED
31. Substituted 2.0 with -1.0 → KILLED
32. Substituted 2.0 with -2.0 → KILLED
33. Substituted 2.0 with -2.0 → KILLED
34. Substituted 2.0 with 3.0 → KILLED
35. Substituted 2.0 with 3.0 → KILLED
36. Substituted 2.0 with 1.0 → KILLED
37. Substituted 2.0 with 1.0 → KILLED
38. Incremented (a++) double local variable number 7 → SURVIVED
39. Incremented (a++) double local variable number 9 → SURVIVED
40. 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

1. Negated double local variable number 7 → KILLED
2. Incremented (a++) double local variable number 7 → KILLED

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

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

2. replaced return value with null for org/jfree/data/Range::expand → KILLED
3. mutated return of Object value for org/jfree/data/Range::expand to (if (x != null) null else throw new RuntimeException) → KILLED
4. Negated double local variable number 7 → KILLED
5. Negated double local variable number 9 → KILLED
6. Incremented (a++) double local variable number 7 → SURVIVED
7. Incremented (a++) double local variable number 9 → SURVIVED
8. Decremented (a--) double local variable number 7 → SURVIVED
9. 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

1. replaced call to org/jfree/data/Range::shift with argument → KILLED
2. Substituted 0 with 1 → SURVIVED
3. removed call to org/jfree/data/Range::shift → KILLED
4. replaced return value with null for org/jfree/data/Range::shift → KILLED
5. mutated return of Object value for org/jfree/data/Range::shift to (if (x != null) null else throw new RuntimeException) → KILLED
6. Negated double local variable number 1 → KILLED

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

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

1. negated conditional → KILLED
2. removed conditional - replaced equality check with false → KILLED
3. removed conditional - replaced equality check with true → KILLED
4. Negated integer local variable number 3 → SURVIVED
5. equal to less than → KILLED
6. equal to less or equal → SURVIVED

366 7. equal to greater than → KILLED
8. equal to greater or equal → KILLED
9. equal to not equal → KILLED
10. Incremented (a++) integer local variable number 3 → SURVIVED
11. Decremented (a--) integer local variable number 3 → SURVIVED
12. Incremented (++a) integer local variable number 3 → KILLED
13. Decremented (--a) integer local variable number 3 → KILLED

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

367 8. Replaced double operation by second member → KILLED
9. Replaced double addition with subtraction → KILLED
10. Replaced double addition with multiplication → KILLED
11. Replaced double addition with division → KILLED
12. Replaced double addition with modulus → KILLED
13. Incremented (a++) double local variable number 1 → KILLED
14. Decremented (a--) double local variable number 1 → KILLED
15. Incremented (++a) double local variable number 1 → KILLED
16. Decremented (--a) double local variable number 1 → KILLED

368 1. Replaced double addition with subtraction → KILLED
2. removed call to org/jfree/data/Range::getUpperBound → KILLED
3. Negated double local variable number 1 → KILLED
4. Replaced double operation with first member → KILLED
5. Replaced double operation by second member → KILLED

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

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

1. changed conditional boundary → SURVIVED
2. Substituted 0.0 with 1.0 → SURVIVED
3. negated conditional → KILLED
4. removed conditional - replaced comparison check with false → KILLED
5. removed conditional - replaced comparison check with true → SURVIVED
6. Negated double local variable number 0 → KILLED
7. Substituted 0.0 with 1.0 → SURVIVED
8. Substituted 0.0 with -1.0 → SURVIVED
9. Substituted 0.0 with 1.0 → SURVIVED

390

10. Substituted 0.0 with -1.0 → SURVIVED
11. greater or equal to less than → KILLED
12. greater or equal to less or equal → KILLED
13. greater or equal to greater than → SURVIVED
14. greater or equal to equal → SURVIVED
15. greater or equal to not equal → KILLED
16. Incremented (a++) double local variable number 0 → KILLED
17. 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

1. replaced call to java/lang/Math::min with argument → KILLED
2. Substituted 0.0 with 1.0 → KILLED
3. Replaced double addition with subtraction → KILLED
4. removed call to java/lang/Math::min → KILLED
5. replaced double return with 0.0d for org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
6. replaced return of double value with -(x + 1) for org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
7. Negated double local variable number 0 → KILLED
8. Negated double local variable number 2 → KILLED
9. Replaced double operation with first member → KILLED
10. Replaced double operation by second member → KILLED
11. Replaced double addition with subtraction → KILLED
12. Replaced double addition with multiplication → KILLED

391

13. Replaced double addition with division → KILLED
14. Replaced double addition with modulus → KILLED
15. Substituted 0.0 with 1.0 → KILLED
16. Substituted 0.0 with -1.0 → KILLED
17. Substituted 0.0 with 1.0 → KILLED
18. Substituted 0.0 with -1.0 → KILLED
19. Incremented (a++) double local variable number 0 → SURVIVED
20. Incremented (a++) double local variable number 2 → SURVIVED
21. 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

394

1. Replaced double addition with subtraction → SURVIVED
2. replaced double return with 0.0d for org/jfree/data/Range::shiftWithNoZeroCrossing → SURVIVED
3. replaced return of double value with -(x + 1) for org/jfree/data/Range::shiftWithNoZeroCrossing → SURVIVED
4. Negated double local variable number 0 → SURVIVED
5. Negated double local variable number 2 → SURVIVED

6. Replaced double operation with first member → SURVIVED
7. Replaced double operation by second member → SURVIVED
8. Replaced double addition with subtraction → SURVIVED
9. Replaced double addition with multiplication → SURVIVED
10. Replaced double addition with division → SURVIVED
11. Replaced double addition with modulus → SURVIVED
12. Incremented (a++) double local variable number 0 → SURVIVED
13. Incremented (a++) double local variable number 2 → SURVIVED
14. Decremented (a--) double local variable number 0 → SURVIVED
15. Decremented (a--) double local variable number 2 → SURVIVED
16. Incremented (++a) double local variable number 0 → SURVIVED
17. Incremented (++a) double local variable number 2 → SURVIVED
18. Decremented (--a) double local variable number 0 → SURVIVED
19. Decremented (--a) double local variable number 2 → SURVIVED

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

1. changed conditional boundary → SURVIVED
2. Substituted 0.0 with 1.0 → KILLED
3. negated conditional → KILLED
4. removed conditional - replaced comparison check with false → SURVIVED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → KILLED
7. Substituted 0.0 with 1.0 → KILLED
8. Substituted 0.0 with -1.0 → SURVIVED
9. Substituted 0.0 with 1.0 → KILLED

410 10. Substituted 0.0 with -1.0 → SURVIVED
11. greater or equal to less than → KILLED
12. greater or equal to less or equal → KILLED
13. greater or equal to greater than → SURVIVED
14. greater or equal to equal → KILLED
15. greater or equal to not equal → SURVIVED
16. Incremented (a++) double local variable number 1 → KILLED
17. Decremented (a--) double local variable number 1 → KILLED
18. Incremented (++a) double local variable number 1 → KILLED
19. Decremented (--a) double local variable number 1 → KILLED

411 1. removed call to java/lang/IllegalArgumentException::<init> → SURVIVED

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

413 8. Replaced double operation by second member → KILLED
9. Replaced double multiplication with division → KILLED
10. Replaced double multiplication with modulus → KILLED
11. Replaced double multiplication with addition → KILLED
12. Replaced double multiplication with subtraction → KILLED
13. Incremented (a++) double local variable number 1 → KILLED
14. Decremented (a--) double local variable number 1 → KILLED
15. Incremented (++a) double local variable number 1 → KILLED
16. Decremented (--a) double local variable number 1 → KILLED

414 1. Replaced double multiplication with division → KILLED
2. removed call to org/jfree/data/Range::getUpperBound → KILLED
3. Negated double local variable number 1 → KILLED
4. Replaced double operation with first member → KILLED
5. Replaced double operation by second member → KILLED
6. Replaced double multiplication with division → KILLED
7. Replaced double multiplication with modulus → KILLED
8. Replaced double multiplication with addition → KILLED
9. Replaced double multiplication with subtraction → KILLED
10. Incremented (a++) double local variable number 1 → SURVIVED
11. Decremented (a--) double local variable number 1 → SURVIVED
12. Incremented (++a) double local variable number 1 → KILLED

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

7. Substituted 0 with -1 → KILLED

1. replaced boolean return with false for org/jfree/data/Range::equals → KILLED
2. Substituted 1 with 0 → KILLED
3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
4. Substituted 1 with 0 → KILLED

436

5. Substituted 1 with -1 → SURVIVED
6. Substituted 1 with -1 → SURVIVED
7. Substituted 1 with 2 → KILLED
8. Substituted 1 with 0 → KILLED

1. replaced boolean return with false for org/jfree/data/Range::isNaNRange → KILLED
2. replaced boolean return with true for org/jfree/data/Range::isNaNRange → KILLED
3. Substituted 1 with 0 → KILLED
4. Substituted 0 with 1 → KILLED
5. negated conditional → KILLED
6. negated conditional → KILLED
7. removed call to java/lang/Double::isNaN → KILLED
8. removed call to java/lang/Double::isNaN → KILLED
9. removed conditional - replaced equality check with false → KILLED
10. removed conditional - replaced equality check with false → KILLED
11. removed conditional - replaced equality check with true → KILLED
12. removed conditional - replaced equality check with true → KILLED
13. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
15. Negated double field lower → SURVIVED
16. Negated double field upper → SURVIVED
17. Substituted 0 with 1 → KILLED
18. Substituted 1 with 0 → KILLED
19. Substituted 1 with -1 → SURVIVED
20. Substituted 0 with -1 → KILLED

448

21. Substituted 1 with -1 → SURVIVED
22. Substituted 1 with 2 → KILLED
23. Substituted 0 with 1 → KILLED
24. Substituted 1 with 0 → KILLED
25. Substituted 0 with -1 → KILLED
26. equal to less than → KILLED
27. equal to less than → KILLED
28. equal to less or equal → SURVIVED
29. equal to less or equal → SURVIVED
30. equal to greater than → KILLED
31. equal to greater than → KILLED
32. equal to greater or equal → KILLED
33. equal to greater or equal → KILLED
34. equal to not equal → KILLED
35. equal to not equal → KILLED
36. Incremented (a++) double field lower → KILLED
37. Incremented (a++) double field upper → SURVIVED
38. Decremented (a--) double field lower → KILLED
39. Decremented (a--) double field upper → SURVIVED
40. Incremented (++a) double field lower → KILLED
41. Incremented (++a) double field upper → SURVIVED
42. Decremented (--a) double fieldlower → KILLED
43. Decremented (--a) double fieldupper → SURVIVED

1. removed call to java/lang/Double::doubleToLongBits → SURVIVED
2. Negated double field lower → SURVIVED
3. Incremented (a++) double field lower → SURVIVED

460

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

461

1. Substituted 32 with 33 → SURVIVED
2. Replaced Unsigned Shift Right with Shift Left → SURVIVED
3. Replaced XOR with AND → SURVIVED
4. Negated long local variable number 2 → SURVIVED
5. Negated long local variable number 2 → SURVIVED
6. Substituted 32 with 1 → SURVIVED

7. Substituted 32 with 0 → SURVIVED
8. Substituted 32 with -1 → SURVIVED
9. Substituted 32 with -32 → SURVIVED
10. Substituted 32 with 33 → SURVIVED
11. Substituted 32 with 31 → SURVIVED
12. Incremented (a++) long local variable number 1 → SURVIVED
13. Incremented (a++) long local variable number 1 → SURVIVED
14. Decremented (a--) long local variable number 2 → SURVIVED
15. Decremented (a--) long local variable number 2 → SURVIVED
16. Incremented (++a) long local variable number 2 → SURVIVED
17. Incremented (++a) long local variable number 2 → SURVIVED
18. Decremented (--a) long local variable number 1 → SURVIVED
19. Decremented (--a) long local variable number 1 → SURVIVED

462
1. removed call to java/lang/Double::doubleToLongBits → SURVIVED
2. Negated double field upper → SURVIVED
3. Incremented (a++) double field upper → SURVIVED
4. Decremented (a--) double field upper → SURVIVED
5. Incremented (++a) double field upper → SURVIVED
6. Decremented (--a) double fieldupper → SURVIVED

463
1. Substituted 29 with 30 → SURVIVED
2. Substituted 32 with 33 → SURVIVED
3. Replaced integer multiplication with division → SURVIVED
4. Replaced Unsigned Shift Right with Shift Left → SURVIVED
5. Replaced XOR with AND → SURVIVED
6. Replaced integer addition with subtraction → SURVIVED
7. Negated integer local variable number 1 → SURVIVED
8. Negated long local variable number 2 → SURVIVED
9. Negated long local variable number 2 → SURVIVED
10. Replaced integer operation with first member → SURVIVED
11. Replaced integer operation with first member → SURVIVED
12. Replaced integer operation by second member → SURVIVED
13. Replaced integer operation by second member → SURVIVED
14. Replaced integer multiplication with division → SURVIVED
15. Replaced integer addition with subtraction → SURVIVED
16. Replaced integer multiplication with modulus → SURVIVED
17. Replaced integer addition with multiplication → SURVIVED
18. Replaced integer multiplication with addition → SURVIVED
19. Replaced integer addition with division → SURVIVED
20. Replaced integer multiplication with subtraction → SURVIVED
21. Replaced integer addition with modulus → SURVIVED
22. Substituted 29 with 1 → SURVIVED
23. Substituted 32 with 1 → SURVIVED
24. Substituted 29 with 0 → SURVIVED
25. Substituted 32 with 0 → SURVIVED
26. Substituted 29 with -1 → SURVIVED
27. Substituted 32 with -1 → SURVIVED
28. Substituted 29 with -29 → SURVIVED
29. Substituted 32 with -32 → SURVIVED
30. Substituted 29 with 30 → SURVIVED
31. Substituted 32 with 33 → SURVIVED
32. Substituted 29 with 28 → SURVIVED
33. Substituted 32 with 31 → SURVIVED
34. Incremented (a++) integer local variable number 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 → SURVIVED
39. Decremented (a--) long local variable number 2 → SURVIVED
40. Incremented (++a) integer local variable number 1 → SURVIVED
41. Incremented (++a) long local variable number 2 → SURVIVED
42. Incremented (++a) long local variable number 2 → SURVIVED
43. Decremented (--a) integer local variable number 3 → SURVIVED
44. Decremented (--a) long local variable number 1 → SURVIVED
45. Decremented (--a) long local variable number 1 → SURVIVED

464
1. replaced int return with 0 for org/jfree/data/Range::hashCode → SURVIVED
2. replaced return of integer sized value with (x == 0 ? 1 : 0) → SURVIVED

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

Active mutators

- ABS_MUTATOR
- AOD_1_MUTATOR
- AOD_2_MUTATOR
- AOR_1_MUTATOR
- AOR_2_MUTATOR
- AOR_3_MUTATOR
- AOR_4_MUTATOR
- ARGUMENT_PROPAGATION_MUTATOR
- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- CONSTRUCTOR_CALL_MUTATOR
- CRCR_1_MUTATOR
- CRCR_2_MUTATOR
- CRCR_3_MUTATOR
- CRCR_4_MUTATOR
- CRCR_5_MUTATOR
- CRCR_6_MUTATOR
- EMPTY_RETURN_VALUES
- EXPERIMENTAL_BIGINTEGER_MUTATOR
- EXPERIMENTAL_MEMBER_VARIABLE_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_[0-99]
- EXPERIMENTAL_SWITCH_MUTATOR
- INCREMENTS_MUTATOR
- INLINE_CONSTANT_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NAKED_RECEIVER
- NEGATE_CONDITIONALS_MUTATOR
- NON_VOID_METHOD_CALL_MUTATOR

- NULL_RETURN_VALUES
- OBBN_1_MUTATOR
- OBBN_2_MUTATOR
- OBBN_3_MUTATOR
- PRIMITIVE_RETURN_VALS_MUTATOR
- REMOVE_CONDITIONALS_EQUAL_ELSE_MUTATOR
- REMOVE_CONDITIONALS_EQUAL_IF_MUTATOR
- REMOVE_CONDITIONALS_ORDER_ELSE_MUTATOR
- REMOVE_CONDITIONALS_ORDER_IF_MUTATOR
- REMOVE_INCREMENTS_MUTATOR
- RETURN_VALS_MUTATOR
- ROR_1_MUTATOR
- ROR_2_MUTATOR
- ROR_3_MUTATOR
- ROR_4_MUTATOR
- ROR_5_MUTATOR
- UOI_1_MUTATOR
- UOI_2_MUTATOR
- UOI_3_MUTATOR
- UOI_4_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullMedian (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (2 ms)
- org.jfree.chart.axis.DateAxisTest.testHashCode(org.jfree.chart.axis.DateAxisTest) (4 ms)
- org.jfree.chart.XYStepChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.XYStepChartTest) (3 ms)
- org.jfree.chart.axis.PeriodAxisTest.testSerialization(org.jfree.chart.axis.PeriodAxisTest) (9 ms)
- org.jfree.chart.StackedBarChartTest.testReplaceDataset(org.jfree.chart.StackedBarChartTest) (3 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateToFindRangeBounds_MultiValueCategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.ScatterPlotTest.testDrawWithNullInfo(org.jfree.chart.ScatterPlotTest) (6 ms)
- org.jfree.chart.axis.ValueAxisTest.testEquals(org.jfree.chart.axis.ValueAxisTest) (1 ms)
- org.jfree.chart.plot.FastScatterPlotTest.testSerialization(org.jfree.chart.plot.FastScatterPlotTest) (5 ms)
- org.jfree.chart.plot.XYPlotTest.testRangeMarkerIndices(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.chart.plot.CategoryPlotTest.testAxisIndices(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.BarChartTest.testReplaceDataset(org.jfree.chart.BarChartTest) (2 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testBug1572478Horizontal (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (22 ms)
- org.jfree.chart.LineChart3DTest.testSetSeriesURLGenerator(org.jfree.chart.LineChart3DTest) (5 ms)
- org.jfree.chart.plot.CategoryPlotTest.testSerialization2(org.jfree.chart.plot.CategoryPlotTest) (4 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testAddValue (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (2 ms)
- org.jfree.chart.axis.PeriodAxisTest.testCloning(org.jfree.chart.axis.PeriodAxisTest) (1 ms)
- org.jfree.chart.LineChartTest.testReplaceDataset(org.jfree.chart.LineChartTest) (2 ms)
- org.jfree.chart.BarChart3DTest.testSetSeriesURLGenerator(org.jfree.chart.BarChart3DTest) (1 ms)
- org.jfree.data.time.TimePeriodValuesCollectionTest.testGetDomainBoundsWithoutInterval (org.jfree.data.time.TimePeriodValuesCollectionTest) (0 ms)
- org.jfree.chart.renderer.category.BarRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.category.BarRendererTest) (1 ms)
- org.jfree.chart.TimeSeriesChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.TimeSeriesChartTest) (6 ms)
- org.jfree.chart.renderer.category.ScatterRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.ScatterRendererTest) (0 ms)
- org.jfree.chart.renderer.category.BarRendererTest.testGetLegendItem (org.jfree.chart.renderer.category.BarRendererTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testXYAutoRange1(org.jfree.chart.axis.NumberAxisTest) (6 ms)
- org.jfree.chart.XYLineChartTest.testReplaceDataset(org.jfree.chart.XYLineChartTest) (2 ms)
- org.jfree.chart.renderer.category.StatisticalLineAndShapeRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.StatisticalLineAndShapeRendererTest) (0 ms)
- org.jfree.chart.renderer.xy.StandardXYItemRendererTest.testNoDisplayedItem (org.jfree.chart.renderer.xy.StandardXYItemRendererTest) (2 ms)
- org.jfree.chart.axis.CyclicNumberAxisTest.testHashCode(org.jfree.chart.axis.CyclicNumberAxisTest) (0 ms)
- org.jfree.chart.renderer.xy.XYBarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.xy.XYBarRendererTest) (0 ms)

- org.jfree.data.xy.XYSeriesCollectionTest.testGetDomainBounds(org.jfree.data.xy.XYSeriesCollectionTest) (0 ms)
- org.jfree.chart.plot.SpiderWebPlotTest.testDrawWithNullInfo(org.jfree.chart.plot.SpiderWebPlotTest) (5 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoomInBoth(org.jfree.chart.ChartPanelTest) (6 ms)
- org.jfree.chart.ChartPanelTest.test2502355_restoreAutoDomainBounds(org.jfree.chart.ChartPanelTest) (25 ms)
- org.jfree.chart.WaterfallChartTest.testDrawWithNullInfo(org.jfree.chart.WaterfallChartTest) (5 ms)
- org.jfree.chart.TimeSeriesChartTest.testReplaceDataset(org.jfree.chart.TimeSeriesChartTest) (3 ms)
- org.jfree.chart.renderer.xy.XYLineAndShapeRendererTest.testFindDomainBounds (org.jfree.chart.renderer.xy.XYLineAndShapeRendererTest) (1 ms)
- org.jfree.chart.axis.SubCategoryAxisTest.test2275695(org.jfree.chart.axis.SubCategoryAxisTest) (4 ms)
- org.jfree.chart.axis.ValueAxisTest.testAxisMargins(org.jfree.chart.axis.ValueAxisTest) (1 ms)
- org.jfree.chart.plot.CombinedDomainXYPlotTest.testNotification (org.jfree.chart.plot.CombinedDomainXYPlotTest) (9 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds_CategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testGetLegendItem (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (1 ms)
- org.jfree.chart.plot.CategoryPlotTest.testAxisLocationIndices(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.plot.ThermometerPlotTest.testSerialization2(org.jfree.chart.plot.ThermometerPlotTest) (3 ms)
- org.jfree.chart.JFreeChartTest.testSerialization4(org.jfree.chart.JFreeChartTest) (181 ms)
- org.jfree.data.xy.DefaultOHLCDatasetTest.testDataRange(org.jfree.data.xy.DefaultOHLCDatasetTest) (0 ms)
- org.jfree.chart.axis.ColorBarTest.testHashCode(org.jfree.chart.axis.ColorBarTest) (0 ms)
- org.jfree.chart.renderer.xy.CandlestickRendererTest.testFindRangeBounds (org.jfree.chart.renderer.xy.CandlestickRendererTest) (0 ms)
- org.jfree.chart.renderer.xy.StackedXYAreaRendererTest.testFindRangeBounds (org.jfree.chart.renderer.xy.StackedXYAreaRendererTest) (1 ms)
- org.jfree.chart.GanttChartTest.testSetSeriesURLGenerator(org.jfree.chart.GanttChartTest) (1 ms)
- org.jfree.chart.renderer.xy.XYBarRendererTest.testFindDomainBounds2 (org.jfree.chart.renderer.xy.XYBarRendererTest) (0 ms)
- org.jfree.chart.block.GridArrangementTest.testGridNotFull_FR(org.jfree.chart.block.GridArrangementTest) (0 ms)
- org.jfree.chart.renderer.category.StatisticalBarRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.StatisticalBarRendererTest) (4 ms)
- org.jfree.chart.axis.ModuloAxisTest.testCloning(org.jfree.chart.axis.ModuloAxisTest) (0 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest.testAddUpdatesCachedRange (org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest) (0 ms)
- org.jfree.chart.renderer.xy.ClusteredXYBarRendererTest.testFindDomainBounds (org.jfree.chart.renderer.xy.ClusteredXYBarRendererTest) (1 ms)
- org.jfree.chart.plot.PolarPlotTest.testTranslateToJava2D_NumberAxisAndMargin (org.jfree.chart.plot.PolarPlotTest) (1 ms)
- org.jfree.chart.XYAreaChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.XYAreaChartTest) (1 ms)
- org.jfree.chart.AreaChartTest.testDrawWithNullInfo(org.jfree.chart.AreaChartTest) (468 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateToFindRangeBounds2_XYDataset (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.block.RectangleConstraintTest.testCalculateConstrainedSize (org.jfree.chart.block.RectangleConstraintTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindDomainBounds_NaN (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.plot.CategoryPlotTest.testSerialization4(org.jfree.chart.plot.CategoryPlotTest) (10 ms)
- org.jfree.data.RangeTest.testExpand(org.jfree.data.RangeTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testCumulativeRange_NaN(org.jfree.data.general.DatasetUtilitiesTest) (6 ms)
- org.jfree.chart.renderer.category.MinMaxCategoryRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.MinMaxCategoryRendererTest) (5 ms)
- org.jfree.chart.axis.NumberAxis3DTest.testSerialization(org.jfree.chart.axis.NumberAxis3DTest) (8 ms)
- org.jfree.chart.renderer.xy.XYStepRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.xy.XYStepRendererTest) (4 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testGetValue (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (0 ms)
- org.jfree.chart.plot.CombinedDomainCategoryPlotTest.testCloning (org.jfree.chart.plot.CombinedDomainCategoryPlotTest) (1 ms)
- org.jfree.chart.axis.ModuloAxisTest.testEquals(org.jfree.chart.axis.ModuloAxisTest) (1 ms)
- org.jfree.chart.renderer.xy.HighLowRendererTest.testFindRangeBounds (org.jfree.chart.renderer.xy.HighLowRendererTest) (0 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testCloning (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (0 ms)
- org.jfree.chart.axis.ColorBarTest.testCloning(org.jfree.chart.axis.ColorBarTest) (1 ms)

- org.jfree.chart.renderer.xy.XYDotRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYDotRendererTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testBug2849731_2(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.LineChart3DTest.testReplaceDataset(org.jfree.chart.LineChart3DTest) (9 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest.testEquals (org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest) (0 ms)
- org.jfree.chart.renderer.category.StackedBarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.StackedBarRendererTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testSerialization4(org.jfree.chart.plot.XYPlotTest) (13 ms)
- org.jfree.data.RangeTest.testConstrain(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullQ3 (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (6 ms)
- org.jfree.chart.GanttChartTest.testDrawWithNullInfo(org.jfree.chart.GanttChartTest) (8 ms)
- org.jfree.chart.BarChart3DTest.testDrawWithNullInfo(org.jfree.chart.BarChart3DTest) (6 ms)
- org.jfree.chart.renderer.xy.XYBarRendererTest.testFindDomainBounds (org.jfree.chart.renderer.xy.XYBarRendererTest) (0 ms)
- org.jfree.chart.block.BorderArrangementTest.testBugX(org.jfree.chart.block.BorderArrangementTest) (0 ms)
- org.jfree.chart.plot.MeterIntervalTest.testEquals(org.jfree.chart.plot.MeterIntervalTest) (0 ms)
- org.jfree.data.RangeTest.testCombine(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.XYStepAreaChartTest.testReplaceDataset(org.jfree.chart.XYStepAreaChartTest) (5 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateDomainBounds_NaN2 (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testGetRangeAxisForDataset(org.jfree.chart.plot.XYPlotTest) (2 ms)
- org.jfree.chart.axis.LogAxisTest.testAutoRange1(org.jfree.chart.axis.LogAxisTest) (3 ms)
- org.jfree.chart.axis.NumberAxisTest.testCloning(org.jfree.chart.axis.NumberAxisTest) (0 ms)
- org.jfree.chart.BarChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.BarChartTest) (2 ms)
- org.jfree.chart.plot.XYPlotTest.testRendererIndices(org.jfree.chart.plot.XYPlotTest) (0 ms)
- org.jfree.chart.BarChart3DTest.testReplaceDataset(org.jfree.chart.BarChart3DTest) (2 ms)
- org.jfree.chart.block.GridArrangementTest.testNullBlock_FR(org.jfree.chart.block.GridArrangementTest) (1 ms)
- org.jfree.data.RangeTest.testHashCode(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.renderer.category.StatisticalBarRendererTest.testDrawWithNullMeanVertical (org.jfree.chart.renderer.category.StatisticalBarRendererTest) (34 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindStackedRangeBounds_CategoryDataset3 (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.plot.FastScatterPlotTest.testDrawWithNullInfo(org.jfree.chart.plot.FastScatterPlotTest) (8 ms)
- org.jfree.chart.plot.CombinedDomainCategoryPlotTest.testNotification (org.jfree.chart.plot.CombinedDomainCategoryPlotTest) (7 ms)
- org.jfree.chart.plot.MeterPlotTest.testCloning(org.jfree.chart.plot.MeterPlotTest) (1 ms)
- org.jfree.chart.axis.SymbolAxisTest.testEquals(org.jfree.chart.axis.SymbolAxisTest) (2 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindStackedRangeBoundsForTableXYDataset2 (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.StackedAreaChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.StackedAreaChartTest) (2 ms)
- org.jfree.chart.renderer.xy.StandardXYItemRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.StandardXYItemRendererTest) (1 ms)
- org.jfree.chart.axis.DateAxisTest.testSerialization(org.jfree.chart.axis.DateAxisTest) (46 ms)
- org.jfree.chart.plot.CategoryPlotTest.testRendererIndices(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testEquals_ObjectList3(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.renderer.xy.XYBarRendererTest.testFindRangeBounds2 (org.jfree.chart.renderer.xy.XYBarRendererTest) (1 ms)
- org.jfree.chart.renderer.category.StatisticalLineAndShapeRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.StatisticalLineAndShapeRendererTest) (2 ms)
- org.jfree.chart.plot.CombinedDomainXYPlotTest.testSerialization (org.jfree.chart.plot.CombinedDomainXYPlotTest) (24 ms)
- org.jfree.chart.plot.XYPlotTest.testCloneIndependence(org.jfree.chart.plot.XYPlotTest) (2 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindDomainBounds(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testSetRange(org.jfree.chart.axis.DateAxisTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testCloning(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoom(org.jfree.chart.ChartPanelTest) (5 ms)
- org.jfree.chart.axis.DateAxisTest.testSetMinimumDate(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindMinimumDomainValue (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testBug2849731(org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.data.RangeTest.testIsNaNRange(org.jfree.data.RangeTest) (0 ms)

- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateSecondB(org.jfree.chart.axis.DateAxisTest) (3 ms)
- org.jfree.chart.axis.ValueAxisTest.test3555275(org.jfree.chart.axis.ValueAxisTest) (6 ms)
- org.jfree.chart.plot.XYPlotTest.testSerialization5(org.jfree.chart.plot.XYPlotTest) (16 ms)
- org.jfree.chart.XYAreaChartTest.testDrawWithNullInfo(org.jfree.chart.XYAreaChartTest) (11 ms)
- org.jfree.chart.BarChartTest.testDrawWithNullInfo(org.jfree.chart.BarChartTest) (5 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullMaxOutlier (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (4 ms)
- org.jfree.chart.plot.ThermometerPlotTest.testEquals(org.jfree.chart.plot.ThermometerPlotTest) (1 ms)
- org.jfree.chart.XYStepAreaChartTest.testDrawWithNullInfo(org.jfree.chart.XYStepAreaChartTest) (2 ms)
- org.jfree.chart.renderer.category.StatisticalBarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.StatisticalBarRendererTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testEquals(org.jfree.chart.axis.NumberAxisTest) (1 ms)
- org.jfree.data.RangeTest.testCombineIgnoringNaN(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.StackedBarChart3DTest.testDrawWithNullInfo(org.jfree.chart.StackedBarChart3DTest) (4 ms)
- org.jfree.chart.plot.CategoryPlotTest.testSerialization5(org.jfree.chart.plot.CategoryPlotTest) (10 ms)
- org.jfree.chart.XYLineChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.XYLineChartTest) (1 ms)
- org.jfree.chart.plot.CategoryPlotTest.testMapDatasetToDomainAxis(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testGetRangeAxisForDataset(org.jfree.chart.plot.CategoryPlotTest) (2 ms)
- org.jfree.chart.XYAreaChartTest.testReplaceDataset(org.jfree.chart.XYAreaChartTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds3(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.test1654215(org.jfree.chart.plot.CategoryPlotTest) (5 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullMinOutlier (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (2 ms)
- org.jfree.data.time.DateRangeTest.testSerialization(org.jfree.data.time.DateRangeTest) (0 ms)
- org.jfree.chart.plot.FastScatterPlotTest.testEquals(org.jfree.chart.plot.FastScatterPlotTest) (1 ms)
- org.jfree.chart.axis.LogarithmicAxisTest.testSerialization(org.jfree.chart.axis.LogarithmicAxisTest) (3 ms)
- org.jfree.chart.plot.XYPlotTest.testAxisIndices(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.data.time.TimeSeriesCollectionTest.testGetRangeBounds2 (org.jfree.data.time.TimeSeriesCollectionTest) (1 ms)
- org.jfree.chart.plot.CombinedDomainXYPlotTest.testRemoveSubplot (org.jfree.chart.plot.CombinedDomainXYPlotTest) (1 ms)
- org.jfree.chart.annotations.XYTitleAnnotationTest.testDrawWithNullInfo (org.jfree.chart.annotations.XYTitleAnnotationTest) (13 ms)
- org.jfree.chart.renderer.category.IntervalBarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.IntervalBarRendererTest) (1 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateYearB(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testSerialization1(org.jfree.chart.plot.XYPlotTest) (6 ms)
- org.jfree.chart.axis.DateAxisTest.testEquals(org.jfree.chart.axis.DateAxisTest) (95 ms)
- org.jfree.chart.axis.DateAxisTest.testBug2201869(org.jfree.chart.axis.DateAxisTest) (20 ms)
- org.jfree.chart.axis.DateAxisTest.test1472942(org.jfree.chart.axis.DateAxisTest) (4 ms)
- org.jfree.chart.plot.XYPlotTest.testGetRendererForDataset2(org.jfree.chart.plot.XYPlotTest) (0 ms)
- org.jfree.chart.axis.LogAxisTest.testAutoRange3(org.jfree.chart.axis.LogAxisTest) (1 ms)
- org.jfree.chart.renderer.category.LevelRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.category.LevelRendererTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testCumulativeRange3(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds2(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoomOutRange(org.jfree.chart.ChartPanelTest) (4 ms)
- org.jfree.chart.plot.XYPlotTest.testEquals_ObjectList(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.chart.axis.NumberAxisTest.testAutoRange2(org.jfree.chart.axis.NumberAxisTest) (1 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoomOutDomain(org.jfree.chart.ChartPanelTest) (3 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateDomainBounds_NaN (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.renderer.category.StatisticalBarRendererTest.testDrawWithNullMeanHorizontal (org.jfree.chart.renderer.category.StatisticalBarRendererTest) (10 ms)
- org.jfree.chart.renderer.xy.XYBoxAndWhiskerRendererTest.test2909215 (org.jfree.chart.renderer.xy.XYBoxAndWhiskerRendererTest) (3 ms)
- org.jfree.chart.renderer.category.LineAndShapeRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.category.LineAndShapeRendererTest) (0 ms)
- org.jfree.chart.axis.CyclicNumberAxisTest.testCloning(org.jfree.chart.axis.CyclicNumberAxisTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testDatasetIndices(org.jfree.chart.plot.XYPlotTest) (0 ms)
- org.jfree.chart.plot.ThermometerPlotTest.testSetUnits(org.jfree.chart.plot.ThermometerPlotTest) (0 ms)
- org.jfree.chart.XYBarChartTest.testReplaceDataset(org.jfree.chart.XYBarChartTest) (1 ms)
- org.jfree.chart.axis.LogAxisTest.testHashCode(org.jfree.chart.axis.LogAxisTest) (1 ms)

- org.jfree.chart.renderer.category.CategoryStepRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.category.CategoryStepRendererTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateToFindRangeBounds_StatisticalCategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullMaxRegular (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (3 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds3_CategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.title.PaintScaleLegendTest.testEquals(org.jfree.chart.title.PaintScaleLegendTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testEquals(org.jfree.chart.plot.CategoryPlotTest) (31 ms)
- org.jfree.chart.renderer.category.StatisticalBarRendererTest.testDrawWithNullDeviationVertical (org.jfree.chart.renderer.category.StatisticalBarRendererTest) (13 ms)
- org.jfree.chart.axis.PeriodAxisTest.test2490803(org.jfree.chart.axis.PeriodAxisTest) (1 ms)
- org.jfree.chart.GanttChartTest.testReplaceDataset(org.jfree.chart.GanttChartTest) (2 ms)
- org.jfree.chart.StackedAreaChartTest.testDrawWithNullInfo(org.jfree.chart.StackedAreaChartTest) (4 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testGetValue2 (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (0 ms)
- org.jfree.chart.axis.ModuloAxisTest.testHashCode(org.jfree.chart.axis.ModuloAxisTest) (1 ms)
- org.jfree.chart.plot.CombinedDomainXYPlotTest.testCloning(org.jfree.chart.plot.CombinedDomainXYPlotTest) (2 ms)
- org.jfree.chart.renderer.category.StackedAreaRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.StackedAreaRendererTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindRangeBounds(org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateYearA(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.chart.plot.PolarPlotTest.testEquals(org.jfree.chart.plot.PolarPlotTest) (1 ms)
- org.jfree.chart.axis.PeriodAxisTest.testHashCode(org.jfree.chart.axis.PeriodAxisTest) (3 ms)
- org.jfree.chart.renderer.xy.StackedXYBarRendererTest.testFindDomainBounds (org.jfree.chart.renderer.xy.StackedXYBarRendererTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds2_CategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testMapDatasetToRangeAxis(org.jfree.chart.plot.CategoryPlotTest) (1 ms)
- org.jfree.chart.renderer.xy.XYLineAndShapeRendererTest.testFindRangeBounds (org.jfree.chart.renderer.xy.XYLineAndShapeRendererTest) (0 ms)
- org.jfree.data.time.DateRangeTest.testEquals(org.jfree.data.time.DateRangeTest) (0 ms)
- org.jfree.chart.axis.LogarithmicAxisTest.testJava2DToValue(org.jfree.chart.axis.LogarithmicAxisTest) (1 ms)
- org.jfree.chart.LineChart3DTest.testSetSeriesToolTipGenerator(org.jfree.chart.LineChart3DTest) (2 ms)
- org.jfree.chart.plot.CombinedRangeCategoryPlotTest.testEquals (org.jfree.chart.plot.CombinedRangeCategoryPlotTest) (4 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateSecondA(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.chart.plot.CombinedRangeCategoryPlotTest.testRemoveSubplot (org.jfree.chart.plot.CombinedRangeCategoryPlotTest) (1 ms)
- org.jfree.chart.AreaChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.AreaChartTest) (1 ms)
- org.jfree.data.xy.XYSeriesCollectionTest.testBug3445507(org.jfree.data.xy.XYSeriesCollectionTest) (2 ms)
- org.jfree.chart.LineChartTest.testSetSeriesURLGenerator(org.jfree.chart.LineChartTest) (2 ms)
- org.jfree.data.time.TimeSeriesCollectionTest.testGetRangeBounds(org.jfree.data.time.TimeSeriesCollectionTest) (0 ms)
- org.jfree.chart.axis.LogAxisTest.testXYAutoRange2(org.jfree.chart.axis.LogAxisTest) (3 ms)
- org.jfree.chart.plot.CategoryPlotTest.testDatasetIndices(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindStackedRangeBoundsForTableXYDataset1 (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullMinRegular (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (3 ms)
- org.jfree.chart.StackedAreaChartTest.testSetSeriesURLGenerator(org.jfree.chart.StackedAreaChartTest) (1 ms)
- org.jfree.chart.axis.NumberAxisTest.testHashCode(org.jfree.chart.axis.NumberAxisTest) (1 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest.testSerialization (org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest) (9 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindDomainBounds3(org.jfree.data.general.DatasetUtilitiesTest) (4 ms)
- org.jfree.chart.XYStepChartTest.testDrawWithNullInfo(org.jfree.chart.XYStepChartTest) (4 ms)
- org.jfree.chart.title.PaintScaleLegendTest.testCloning(org.jfree.chart.title.PaintScaleLegendTest) (1 ms)
- org.jfree.data.time.TimeSeriesCollectionTest.testBug3445507(org.jfree.data.time.TimeSeriesCollectionTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testTranslateJava2DToValue(org.jfree.chart.axis.NumberAxisTest) (0 ms)
- org.jfree.chart.XYStepChartTest.testReplaceDataset(org.jfree.chart.XYStepChartTest) (1 ms)
- org.jfree.chart.plot.CombinedRangeCategoryPlotTest.testNotification (org.jfree.chart.plot.CombinedRangeCategoryPlotTest) (18 ms)

- org.jfree.chart.title.PaintScaleLegendTest.testSerialization(org.jfree.chart.title.PaintScaleLegendTest) (4 ms)
- org.jfree.chart.AreaChartTest.testSetSeriesURLGenerator(org.jfree.chart.AreaChartTest) (2 ms)
- org.jfree.chart.WaterfallChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.WaterfallChartTest) (2 ms)
- org.jfree.chart.axis.ModuloAxisTest.testSerialization(org.jfree.chart.axis.ModuloAxisTest) (4 ms)
- org.jfree.chart.axis.LogAxisTest.testCloning(org.jfree.chart.axis.LogAxisTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testMapDatasetToRangeAxis(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testStackedRangeWithMap(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.data.time.DateRangeTest.testClone(org.jfree.data.time.DateRangeTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testAutoRange4(org.jfree.chart.axis.NumberAxisTest) (0 ms)
- org.jfree.chart.plot.CombinedRangeXYPlotTest.testEquals(org.jfree.chart.plot.CombinedRangeXYPlotTest) (1 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullQ1 (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (40 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testCumulativeRange1(org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.BarChart3DTest.testSetSeriesToolTipGenerator(org.jfree.chart.BarChart3DTest) (2 ms)
- org.jfree.chart.plot.XYPlotTest.testSerialization2(org.jfree.chart.plot.XYPlotTest) (12 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (3 ms)
- org.jfree.chart.plot.PolarPlotTest.testCloning(org.jfree.chart.plot.PolarPlotTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindMaximumDomainValue (org.jfree.data.general.DatasetUtilitiesTest) (2 ms)
- org.jfree.chart.plot.CategoryPlotTest.test1169972(org.jfree.chart.plot.CategoryPlotTest) (1 ms)
- org.jfree.chart.renderer.xy.StackedXYAreaRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.xy.StackedXYAreaRendererTest) (7 ms)
- org.jfree.chart.XYStepAreaChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.XYStepAreaChartTest) (0 ms)
- org.jfree.chart.XYBarChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.XYBarChartTest) (3 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest.testGetRangeBounds (org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testSerialization3(org.jfree.chart.plot.CategoryPlotTest) (14 ms)
- org.jfree.chart.ChartPanelTest.test2502355_restoreAutoRangeBounds(org.jfree.chart.ChartPanelTest) (7 ms)
- org.jfree.chart.plot.PiePlot3DTest.testDrawWithNullDataset(org.jfree.chart.plot.PiePlot3DTest) (2 ms)
- org.jfree.data.time.TimeSeriesCollectionTest.testFindDomainBounds (org.jfree.data.time.TimeSeriesCollectionTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindRangeBounds_CategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (2 ms)
- org.jfree.chart.plot.CategoryPlotTest.testGetRendererForDataset2(org.jfree.chart.plot.CategoryPlotTest) (1 ms)
- org.jfree.data.time.TimeSeriesTest.testFindValueRange2(org.jfree.data.time.TimeSeriesTest) (1 ms)
- org.jfree.chart.ChartPanelTest.test2502355_restoreAutoBounds(org.jfree.chart.ChartPanelTest) (62 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testGetRowCount (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (0 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest.testCloning (org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest) (1 ms)
- org.jfree.chart.ScatterPlotTest.testReplaceDataset(org.jfree.chart.ScatterPlotTest) (2 ms)
- org.jfree.chart.renderer.AbstractRendererTest.testOutlinePaintLookup (org.jfree.chart.renderer.AbstractRendererTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testGetDomainAxisForDataset(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.MeterChartTest.testDrawWithNullInfo(org.jfree.chart.MeterChartTest) (21 ms)
- org.jfree.chart.GanttChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.GanttChartTest) (3 ms)
- org.jfree.chart.XYBarChartTest.testDrawWithNullInfo(org.jfree.chart.XYBarChartTest) (14 ms)
- org.jfree.chart.plot.CombinedRangeCategoryPlotTest.testCloning (org.jfree.chart.plot.CombinedRangeCategoryPlotTest) (1 ms)
- org.jfree.chart.renderer.xy.StackedXYAreaRendererTest.testBug1593156 (org.jfree.chart.renderer.xy.StackedXYAreaRendererTest) (3 ms)
- org.jfree.chart.axis.LogAxisTest.testSerialization(org.jfree.chart.axis.LogAxisTest) (13 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoomInDomain(org.jfree.chart.ChartPanelTest) (4 ms)
- org.jfree.chart.renderer.AbstractRendererTest.testFillPaintLookup(org.jfree.chart.renderer.AbstractRendererTest) (0 ms)
- org.jfree.chart.ChartPanelTest.testSetMouseWheelEnabled(org.jfree.chart.ChartPanelTest) (7 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest.testRemove (org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateToFindRangeBounds_BoxAndWhiskerXYDataset (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.BarChartTest.testSetSeriesURLGenerator(org.jfree.chart.BarChartTest) (2 ms)
- org.jfree.chart.JFreeChartTest.testSerialization3(org.jfree.chart.JFreeChartTest) (48 ms)

- org.jfree.chart.plot.MeterIntervalTest.testCloning(org.jfree.chart.plot.MeterIntervalTest) (0 ms)
- org.jfree.chart.renderer.category.StackedBarRenderer3DTest.testFindRangeBounds (org.jfree.chart.renderer.category.StackedBarRenderer3DTest) (1 ms)
- org.jfree.chart.StackedBarChart3DTest.testReplaceDataset(org.jfree.chart.StackedBarChart3DTest) (2 ms)
- org.jfree.chart.plot.XYPlotTest.testCloning3(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.chart.plot.CombinedDomainCategoryPlotTest.testEquals (org.jfree.chart.plot.CombinedDomainCategoryPlotTest) (3 ms)
- org.jfree.chart.plot.XYPlotTest.testCloning2(org.jfree.chart.plot.XYPlotTest) (3 ms)
- org.jfree.chart.LineChart3DTest.testDrawWithNullInfo(org.jfree.chart.LineChart3DTest) (11 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest.testGetRangeBounds (org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateMillisecondB(org.jfree.chart.axis.DateAxisTest) (4 ms)
- org.jfree.chart.plot.CategoryPlotTest.testGetRangeAxisIndex(org.jfree.chart.plot.CategoryPlotTest) (2 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testGetColumnCount (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (0 ms)
- org.jfree.chart.block.GridArrangementTest.testRN(org.jfree.chart.block.GridArrangementTest) (1 ms)
- org.jfree.chart.plot.MeterPlotTest.testSerialization2(org.jfree.chart.plot.MeterPlotTest) (2 ms)
- org.jfree.chart.plot.CombinedRangeXYPlotTest.testNotification(org.jfree.chart.plot.CombinedRangeXYPlotTest) (13 ms)
- org.jfree.chart.LineChartTest.testDrawWithNullInfo(org.jfree.chart.LineChartTest) (5 ms)
- org.jfree.chart.renderer.category.AbstractCategoryItemRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.AbstractCategoryItemRendererTest) (0 ms)
- org.jfree.chart.plot.CombinedRangeCategoryPlotTest.testSerialization (org.jfree.chart.plot.CombinedRangeCategoryPlotTest) (161 ms)
- org.jfree.chart.axis.ColorBarTest.testEquals(org.jfree.chart.axis.ColorBarTest) (1 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateMillisecondA(org.jfree.chart.axis.DateAxisTest) (4 ms)
- org.jfree.chart.StackedBarChart3DTest.testSetSeriesToolTipGenerator(org.jfree.chart.StackedBarChart3DTest) (0 ms)
- org.jfree.chart.plot.CombinedDomainXYPlotTest.testEquals(org.jfree.chart.plot.CombinedDomainXYPlotTest) (3 ms)
- org.jfree.data.RangeTest.testScale(org.jfree.data.RangeTest) (0 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testSerialization (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (1 ms)
- org.jfree.chart.renderer.xy.XYBlockRendererTest.testFindDomainBounds (org.jfree.chart.renderer.xy.XYBlockRendererTest) (0 ms)
- org.jfree.chart.renderer.xy.StackedXYAreaRenderer2Test.testFindRangeBounds (org.jfree.chart.renderer.xy.StackedXYAreaRenderer2Test) (0 ms)
- org.jfree.chart.plot.XYPlotTest.testDrawRangeGridlines(org.jfree.chart.plot.XYPlotTest) (3 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindDomainBounds2(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.GanttChartTest.testDrawWithNullInfo2(org.jfree.chart.GanttChartTest) (30 ms)
- org.jfree.data.RangeTest.testShift(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateDayB(org.jfree.chart.axis.DateAxisTest) (3 ms)
- org.jfree.chart.axis.LogAxisTest.testRefreshTicksWithZeroTickUnit(org.jfree.chart.axis.LogAxisTest) (67 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateToFindRangeBounds1_XYDataset (org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.axis.SymbolAxisTest.testSerialization(org.jfree.chart.axis.SymbolAxisTest) (3 ms)
- org.jfree.chart.axis.ColorBarTest.testSerialization(org.jfree.chart.axis.ColorBarTest) (3 ms)
- org.jfree.data.RangeTest.testEquals(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.renderer.xy.XYDifferenceRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYDifferenceRendererTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testCumulativeRange2(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.data.time.TimePeriodValuesCollectionTest.testGetDomainBoundsWithInterval (org.jfree.data.time.TimePeriodValuesCollectionTest) (0 ms)
- org.jfree.chart.plot.ThermometerPlotTest.testCloning(org.jfree.chart.plot.ThermometerPlotTest) (1 ms)
- org.jfree.chart.renderer.category.GroupedStackedBarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.GroupedStackedBarRendererTest) (0 ms)
- org.jfree.chart.plot.CategoryPlotTest.testRangeMarkerIndices(org.jfree.chart.plot.CategoryPlotTest) (2 ms)
- org.jfree.chart.XYLineChartTest.testDrawWithNullInfo(org.jfree.chart.XYLineChartTest) (5 ms)
- org.jfree.chart.axis.PeriodAxisTest.testEquals(org.jfree.chart.axis.PeriodAxisTest) (0 ms)
- org.jfree.chart.axis.SymbolAxisTest.testCloning(org.jfree.chart.axis.SymbolAxisTest) (1 ms)
- org.jfree.chart.axis.LogAxisTest.testTranslateJava2DToValue(org.jfree.chart.axis.LogAxisTest) (1 ms)
- org.jfree.chart.plot.MeterPlotTest.testEquals(org.jfree.chart.plot.MeterPlotTest) (1 ms)
- org.jfree.chart.plot.ThermometerPlotTest.testSerialization(org.jfree.chart.plot.ThermometerPlotTest) (5 ms)

- org.jfree.chart.axis.CyclicNumberAxisTest.testEquals(org.jfree.chart.axis.CyclicNumberAxisTest) (0 ms)
- org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest.testGetRangeBounds (org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest) (0 ms)
- org.jfree.chart.renderer.xy.XYStepAreaRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.xy.XYStepAreaRendererTest) (8 ms)
- org.jfree.chart.plot.CategoryPlotTest.testGetDomainAxisIndex(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.chart.title.PaintScaleLegendTest.testHashcode(org.jfree.chart.title.PaintScaleLegendTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testBug2849731_3(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testAutoRange3(org.jfree.chart.axis.NumberAxisTest) (1 ms)
- org.jfree.chart.axis.LogAxisTest.testTickMarksVisibleDefault(org.jfree.chart.axis.LogAxisTest) (1 ms)
- org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest.testEquals (org.jfree.data.statistics.DefaultMultiValueCategoryDatasetTest) (1 ms)
- org.jfree.chart.renderer.xy.StackedXYAreaRenderer2Test.testDrawWithEmptyDataset (org.jfree.chart.renderer.xy.StackedXYAreaRenderer2Test) (2 ms)
- org.jfree.data.RangeTest.testContains(org.jfree.data.RangeTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateDayA(org.jfree.chart.axis.DateAxisTest) (3 ms)
- org.jfree.chart.axis.LogAxisTest.testEquals(org.jfree.chart.axis.LogAxisTest) (2 ms)
- org.jfree.chart.plot.CombinedRangeXYPlotTest.testCloning(org.jfree.chart.plot.CombinedRangeXYPlotTest) (3 ms)
- org.jfree.chart.renderer.xy.XYAreaRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.xy.XYAreaRendererTest) (3 ms)
- org.jfree.chart.plot.PolarPlotTest.testTranslateToJava2D_LogAxis(org.jfree.chart.plot.PolarPlotTest) (2 ms)
- org.jfree.chart.axis.LogarithmicAxisTest.testSwitchedLog10(org.jfree.chart.axis.LogarithmicAxisTest) (1 ms)
- org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest.testAdd (org.jfree.data.statistics.DefaultBoxAndWhiskerXYDatasetTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds_IntervalCategoryDataset (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.renderer.xy.XYAreaRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYAreaRendererTest) (1 ms)
- org.jfree.chart.axis.CyclicNumberAxisTest.testSerialization(org.jfree.chart.axis.CyclicNumberAxisTest) (2 ms)
- org.jfree.chart.LineChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.LineChartTest) (3 ms)
- org.jfree.chart.renderer.category.StatisticalBarRendererTest.testDrawWithNullDeviationHorizontal (org.jfree.chart.renderer.category.StatisticalBarRendererTest) (5 ms)
- org.jfree.chart.axis.DateAxisTest.testBug3484403(org.jfree.chart.axis.DateAxisTest) (4 ms)
- org.jfree.chart.plot.MeterIntervalTest.testSerialization(org.jfree.chart.plot.MeterIntervalTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testSetNullRenderer(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.chart.StackedBarChart3DTest.testSetSeriesURLGenerator(org.jfree.chart.StackedBarChart3DTest) (2 ms)
- org.jfree.chart.plot.XYPlotTest.testEquals_ObjectList3(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.chart.renderer.xy.XYStepRendererTest.testDrawWithNullValue (org.jfree.chart.renderer.xy.XYStepRendererTest) (7 ms)
- org.jfree.chart.plot.XYPlotTest.testMapDatasetToDomainAxis(org.jfree.chart.plot.XYPlotTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testSerialization(org.jfree.chart.axis.NumberAxisTest) (4 ms)
- org.jfree.chart.plot.XYPlotTest.testAxisLocationIndices(org.jfree.chart.plot.XYPlotTest) (2 ms)
- org.jfree.data.xy.XYSeriesCollectionTest.testGetRangeBounds(org.jfree.data.xy.XYSeriesCollectionTest) (2 ms)
- org.jfree.chart.axis.PeriodAxisTest.test1932146(org.jfree.chart.axis.PeriodAxisTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds4(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.renderer.xy.XYBarRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYBarRendererTest) (1 ms)
- org.jfree.chart.PieChart3DTest.testNullValueInDataset(org.jfree.chart.PieChart3DTest) (23 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindStackedRangeBounds_CategoryDataset2 (org.jfree.data.general.DatasetUtilitiesTest) (4 ms)
- org.jfree.chart.StackedBarChartTest.testSetSeriesURLGenerator(org.jfree.chart.StackedBarChartTest) (1 ms)
- org.jfree.chart.renderer.category.LevelRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.LevelRendererTest) (5 ms)
- org.jfree.chart.axis.LogAxisTest.testXYAutoRange1(org.jfree.chart.axis.LogAxisTest) (1 ms)
- org.jfree.chart.plot.CategoryPlotTest.testAxisRange(org.jfree.chart.plot.CategoryPlotTest) (1 ms)
- org.jfree.chart.plot.CombinedRangeXYPlotTest.testRemoveSubplot (org.jfree.chart.plot.CombinedRangeXYPlotTest) (2 ms)
- org.jfree.chart.WaterfallChartTest.testSetSeriesURLGenerator(org.jfree.chart.WaterfallChartTest) (1 ms)
- org.jfree.data.RangeTest.testSerialization(org.jfree.data.RangeTest) (1 ms)
- org.jfree.chart.renderer.category.AreaRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.category.AreaRendererTest) (0 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testBug1572478Vertical (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (2 ms)

- org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest.testAdd (org.jfree.data.statistics.DefaultBoxAndWhiskerCategoryDatasetTest) (0 ms)
- org.jfree.chart.plot.CombinedDomainCategoryPlotTest.testSerialization (org.jfree.chart.plot.CombinedDomainCategoryPlotTest) (14 ms)
- org.jfree.chart.block.GridArrangementTest.testNR(org.jfree.chart.block.GridArrangementTest) (1 ms)
- org.jfree.data.time.DateRangeTest.testImmutable(org.jfree.data.time.DateRangeTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testJava2DToValue(org.jfree.chart.axis.DateAxisTest) (5 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindStackedRangeBounds_CategoryDataset1 (org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.block.GridArrangementTest.testRF(org.jfree.chart.block.GridArrangementTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateRangeBounds(org.jfree.data.general.DatasetUtilitiesTest) (0 ms)
- org.jfree.chart.renderer.xy.XYLineAndShapeRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYLineAndShapeRendererTest) (0 ms)
- org.jfree.chart.AreaChartTest.testReplaceDataset(org.jfree.chart.AreaChartTest) (1 ms)
- org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest.testDrawWithNullMean (org.jfree.chart.renderer.category.BoxAndWhiskerRendererTest) (3 ms)
- org.jfree.chart.axis.NumberAxisTest.testXYAutoRange2(org.jfree.chart.axis.NumberAxisTest) (2 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateHourA(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.data.time.TimeSeriesTest.testFindValueRange(org.jfree.data.time.TimeSeriesTest) (0 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoomOutBoth(org.jfree.chart.ChartPanelTest) (4 ms)
- org.jfree.chart.ChartPanelTest.test2502355_zoomInRange(org.jfree.chart.ChartPanelTest) (6 ms)
- org.jfree.chart.axis.LogarithmicAxisTest.testAdjustedLog10(org.jfree.chart.axis.LogarithmicAxisTest) (0 ms)
- org.jfree.chart.plot.FastScatterPlotTest.testEquals2(org.jfree.chart.plot.FastScatterPlotTest) (1 ms)
- org.jfree.chart.plot.CategoryPlotTest.testSerialization(org.jfree.chart.plot.CategoryPlotTest) (5 ms)
- org.jfree.chart.renderer.category.GroupedStackedBarRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.GroupedStackedBarRendererTest) (9 ms)
- org.jfree.chart.plot.XYPlotTest.testCloning4(org.jfree.chart.plot.XYPlotTest) (1 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateMonthA(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.test1654215(org.jfree.chart.plot.XYPlotTest) (4 ms)
- org.jfree.chart.plot.XYPlotTest.testGetDomainAxisForDataset(org.jfree.chart.plot.XYPlotTest) (0 ms)
- org.jfree.chart.StackedBarChartTest.testSetSeriesToolTipGenerator(org.jfree.chart.StackedBarChartTest) (2 ms)
- org.jfree.chart.plot.PiePlotTest.testDrawWithNullLegendLabels(org.jfree.chart.plot.PiePlotTest) (7 ms)
- org.jfree.chart.axis.PeriodAxisTest.testEqualsWithLocale(org.jfree.chart.axis.PeriodAxisTest) (59 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateHourB(org.jfree.chart.axis.DateAxisTest) (3 ms)
- org.jfree.chart.axis.LogAxisTest.testSetLowerBound(org.jfree.chart.axis.LogAxisTest) (0 ms)
- org.jfree.chart.axis.DateAxisTest.testSetMaximumDate(org.jfree.chart.axis.DateAxisTest) (0 ms)
- org.jfree.chart.axis.LogarithmicAxisTest.testValueToJava2D(org.jfree.chart.axis.LogarithmicAxisTest) (0 ms)
- org.jfree.chart.plot.XYPlotTest.testDrawSeriesWithZeroItems(org.jfree.chart.plot.XYPlotTest) (6 ms)
- org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest.testRemove (org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest) (0 ms)
- org.jfree.chart.plot.PolarPlotTest.testTranslateToJava2D_NumberAxis(org.jfree.chart.plot.PolarPlotTest) (2 ms)
- org.jfree.chart.axis.DateAxisTest.testPreviousStandardDateMonthB(org.jfree.chart.axis.DateAxisTest) (1 ms)
- org.jfree.chart.plot.CombinedRangeXYPlotTest.testSerialization (org.jfree.chart.plot.CombinedRangeXYPlotTest) (29 ms)
- org.jfree.data.RangeTest.testIntersects(org.jfree.data.RangeTest) (0 ms)
- org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest.test3072674 (org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest) (0 ms)
- org.jfree.chart.renderer.category.LayeredBarRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.LayeredBarRendererTest) (6 ms)
- org.jfree.chart.plot.CategoryPlotTest.testDomainMarkerIndices(org.jfree.chart.plot.CategoryPlotTest) (0 ms)
- org.jfree.data.RangeTest.testConstructor(org.jfree.data.RangeTest) (0 ms)
- org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest.testGetRangeBounds2 (org.jfree.data.statistics.DefaultStatisticalCategoryDatasetTest) (1 ms)
- org.jfree.chart.StackedBarChartTest.testDrawWithNullInfo(org.jfree.chart.StackedBarChartTest) (4 ms)
- org.jfree.chart.renderer.category.LineAndShapeRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.LineAndShapeRendererTest) (1 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testFindRangeBounds2(org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.renderer.xy.XYBubbleRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYBubbleRendererTest) (1 ms)
- org.jfree.chart.renderer.category.IntervalBarRendererTest.testDrawWithNullInfo (org.jfree.chart.renderer.category.IntervalBarRendererTest) (12 ms)
- org.jfree.chart.renderer.xy.AbstractXYItemRendererTest.testFindDomainBounds (org.jfree.chart.renderer.xy.AbstractXYItemRendererTest) (0 ms)
- org.jfree.chart.TimeSeriesChartTest.testDrawWithNullInfo(org.jfree.chart.TimeSeriesChartTest) (5 ms)

- org.jfree.chart.renderer.category.BarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.category.BarRendererTest) (0 ms)
- org.jfree.chart.axis.NumberAxisTest.testAutoRange1(org.jfree.chart.axis.NumberAxisTest) (0 ms)
- org.jfree.chart.renderer.xy.XYAreaRenderer2Test.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.XYAreaRenderer2Test) (1 ms)
- org.jfree.chart.axis.NumberAxisTest.testSetLowerBound(org.jfree.chart.axis.NumberAxisTest) (0 ms)
- org.jfree.data.general.DatasetUtilitiesTest.testIterateDomainBounds(org.jfree.data.general.DatasetUtilitiesTest) (1 ms)
- org.jfree.chart.StackedAreaChartTest.testReplaceDataset(org.jfree.chart.StackedAreaChartTest) (1 ms)
- org.jfree.chart.plot.XYPlotTest.testEquals(org.jfree.chart.plot.XYPlotTest) (7 ms)
- org.jfree.chart.renderer.AbstractRendererTest.testPaintLookup(org.jfree.chart.renderer.AbstractRendererTest) (1 ms)
- org.jfree.chart.plot.FastScatterPlotTest.testCloning(org.jfree.chart.plot.FastScatterPlotTest) (2 ms)
- org.jfree.chart.renderer.xy.XYAreaRenderer2Test.testDrawWithNullInfo (org.jfree.chart.renderer.xy.XYAreaRenderer2Test) (11 ms)
- org.jfree.chart.block.GridArrangementTest.testRR(org.jfree.chart.block.GridArrangementTest) (2 ms)
- org.jfree.chart.ScatterPlotTest.testSetSeriesToolTipGenerator(org.jfree.chart.ScatterPlotTest) (4 ms)
- org.jfree.chart.plot.XYPlotTest.testDomainMarkerIndices(org.jfree.chart.plot.XYPlotTest) (2 ms)
- org.jfree.chart.renderer.xy.YIntervalRendererTest.testGetLegendItemSeriesIndex (org.jfree.chart.renderer.xy.YIntervalRendererTest) (1 ms)
- org.jfree.chart.axis.ValueAxisTest.testCloning(org.jfree.chart.axis.ValueAxisTest) (1 ms)
- org.jfree.chart.renderer.xy.StackedXYBarRendererTest.testFindRangeBounds (org.jfree.chart.renderer.xy.StackedXYBarRendererTest) (0 ms)
- org.jfree.chart.plot.XYPlotTest.testSerialization3(org.jfree.chart.plot.XYPlotTest) (9 ms)
- org.jfree.chart.plot.MeterPlotTest.testSerialization1(org.jfree.chart.plot.MeterPlotTest) (3 ms)

Report generated by PIT 1.4.11