

Range.java

1	/* =====
	=
2	* JFreeChart : a free chart library for the Java(tm) platform
	*
3	=====
	=
4	*
5	* (C) Copyright 2000-2014, by Object Refinery Limited and Contributors.
6	*
7	* Project Info: http://www.jfree.org/jfreechart/index.html
8	*
9	* This library is free software; you can redistribute it and/or modify it
10	* under the terms of the GNU Lesser General Public License as published by
11	* the Free Software Foundation; either version 2.1 of the License, or
12	* (at your option) any later version.
13	*
14	* This library is distributed in the hope that it will be useful, but
15	* WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
16	* or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public
17	* License for more details.
18	*
19	* You should have received a copy of the GNU Lesser General Public
20	* License along with this library; if not, write to the Free Software

21	* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
22	* USA.
23	*
24	* [Oracle and Java are registered trademarks of Oracle and/or its affiliates.
25	* Other names may be trademarks of their respective owners.]
26	*
27	* -----
28	* Range.java
29	* -----
30	* (C) Copyright 2002-2014, by Object Refinery Limited and Contributors.
31	*
32	* Original Author: David Gilbert (for Object Refinery Limited);
33	* Contributor(s): Chuanhao Chiu;
34	* Bill Kelemen;
35	* Nicolas Brodu;
36	* Sergei Ivanov;
37	*
38	* Changes (from 23-Jun-2001)
39	* -----
40	* 22-Apr-2002 : Version 1, loosely based by code by Bill Kelemen (DG);
41	* 30-Apr-2002 : Added getLength() and getCentralValue() methods. Changed
42	* argument check in constructor (DG);
43	* 13-Jun-2002 : Added contains(double) method (DG);
44	* 22-Aug-2002 : Added fix to combine method where both ranges are null, thanks
45	* to Chuanhao Chiu for reporting and fixing this (DG);

46	* 07-Oct-2002 : Fixed errors reported by Checkstyle (DG);
47	* 26-Mar-2003 : Implemented Serializable (DG);
48	* 14-Aug-2003 : Added equals() method (DG);
49	* 27-Aug-2003 : Added toString() method (BK);
50	* 11-Sep-2003 : Added Clone Support (NB);
51	* 23-Sep-2003 : Fixed Checkstyle issues (DG);
52	* 25-Sep-2003 : Oops, Range immutable, clone not necessary (NB);
53	* 05-May-2004 : Added constrain() and intersects() methods (DG);
54	* 18-May-2004 : Added expand() method (DG);
55	* ----- JFreeChart 1.0.x -----
56	* 11-Jan-2006 : Added new method expandToInclude(Range, double) (DG);
57	* 18-Dec-2007 : New methods intersects(Range) and scale(...) thanks to Sergei
58	* Ivanov (DG);
59	* 08-Jan-2012 : New method combineIgnoringNaN() (DG);
60	* 23-Feb-2014 : Added isNaNRange() method (DG);
61	*
62	*/
63	
64	package org.jfree.data;
65	
66	import java.io.Serializable;
67	import org.jfree.chart.util.ParamChecks;
68	
69	/**
70	* Represents an immutable range of values.
71	*/

72		public strictfp class Range implements Serializable {
73		
74		/** For serialization. */
75		private static final long serialVersionUID = -906333695431863380L;
76		
77		/** The lower bound of the range. */
78		private double lower;
79		
80		/** The upper bound of the range. */
81		private double upper;
82		
83		/**
84		* Creates a new range.
85		*
86		* @param lower the lower bound (must be <= upper bound).
87		* @param upper the upper bound (must be >= lower bound).
88		*/
89		public Range(double lower, double upper) {
90	<u>1</u> <u>9</u>	if (lower > upper) {
91	<u>9</u>	String msg = "Range(double, double): require lower (" + lower
92	<u>1</u> <u>1</u>	+ ") <= upper (" + upper + ").";
93	<u>1</u>	throw new IllegalArgumentException(msg);
94		}
95	<u>6</u>	this.lower = lower;
96	<u>6</u>	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;

11		}
5		
11		
6		
11		/**
7		
11		* Returns the length of the range.
8		
11		*
9		
12		* @return The length.
0		
12		*/
1		
12		public double getLength() {
2		
12	1	return this.upper - this.lower;
3	9	
12		}
4		
12		
5		
12		/**
6		
12		* Returns the central value for the range.
7		
12		*
8		
12		* @return The central value.
9		
13		*/
0		
13		public double getCentralValue() {
1		

13 2	<u>4</u> <u>7</u>	return this.lower / 2.0 + this.upper / 2.0;
13 3		}
13 4		
13 5		/**
13 6		* Returns <code>true</code> if the range contains the specified value and
13 7		* <code>false</code> otherwise.
13 8		*
13 9		* @param value the value to lookup.
14 0		*
14 1		* @return <code>true</code> if the range contains the specified value.
14 2		*/
14 3		public boolean contains(double value) {
14 4	<u>5</u> <u>3</u>	return (value >= this.lower && value <= this.upper);
14 5		}
14 6		
14 7		/**
14 8		* Returns <code>true</code> if the range intersects with the specified

14 9		* range, and <code>false</code> otherwise.
15 0		*
15 1		* @param b0 the lower bound (should be \leq b1).
15 2		* @param b1 the upper bound (should be \geq b0).
15 3		*
15 4		* @return A boolean.
15 5		*/
15 6		public boolean intersects(double b0, double b1) {
15 7	<u>1</u> <u>9</u>	if (b0 \leq this.lower) {
15 8	<u>3</u> <u>4</u>	return (b1 > this.lower);
15 9		}
16 0		else {
16 1	<u>5</u> <u>3</u>	return (b0 < this.upper && b1 \geq b0);
16 2		}
16 3		}
16 4		
16 5		/**

16 6		* Returns <code>true</code> if the range intersects with the specified
16 7		* range, and <code>false</code> otherwise.
16 8		*
16 9		* @param range another range (<code>null</code> not permitted).
17 0		*
17 1		* @return A boolean.
17 2		*
17 3		* @since 1.0.9
17 4		*/
17 5		public boolean intersects(Range range) {
17 6	6	return intersects(range.getLowerBound(), range.getUpperBound());
17 7		}
17 8		
17 9		/**
18 0		* Returns the value within the range that is closest to the specified
18 1		* value.
18 2		*

18 3		* @param value the value.
18 4		*
18 5		* @return The constrained value.
18 6		*/
18 7		public double constrain(double value) {
18 8	<u>5</u>	double result = value;
18 9	<u>1</u> <u>4</u>	if (!contains(value)) {
19 0	<u>1</u> <u>9</u>	if (value > this.upper) {
19 1	<u>5</u>	result = this.upper;
19 2		}
19 3	<u>1</u> <u>9</u>	else if (value < this.lower) {
19 4	<u>5</u>	result = this.lower;
19 5		}
19 6		}
19 7	<u>7</u>	return result;
19 8		}
19 9		

20 0	/**
20 1	* Creates a new range by combining two existing ranges.
20 2	* <P>
20 3	* Note that:
20 4	*
20 5	* either range can be <code>null</code> , in which case the other
20 6	* range is returned;
20 7	* if both ranges are <code>null</code> the return value is
20 8	* <code>null</code> .
20 9	*
21 0	*
21 1	* @param range1 the first range (<code>null</code> permitted).
21 2	* @param range2 the second range (<code>null</code> permitted).
21 3	*
21 4	* @return A new range (possibly <code>null</code>).
21 5	*/
21 6	public static Range combine(Range range1, Range range2) {

21 7	<u>4</u>	if (range1 == null) {
21 8	<u>2</u>	return range2;
21 9		}
22 0	<u>4</u>	if (range2 == null) {
22 1	<u>2</u>	return range1;
22 2		}
22 3	<u>4</u>	double l = Math.min(range1.getLowerBound(), range2.getLowerBound());
22 4	<u>4</u>	double u = Math.max(range1.getUpperBound(), range2.getUpperBound());
22 5	<u>1</u> <u>3</u>	return new Range(l, u);
22 6		}
22 7		
22 8		/**
22 9		* Returns a new range that spans both <code>range1</code> and
23 0		* <code>range2</code> . This method has a special handling to ignore
23 1		* Double.NaN values.
23 2		*
23 3		* @param range1 the first range (<code>null</code> permitted).

23 4		* @param range2 the second range (<code>null</code> permitted).
23 5		*
23 6		* @return A new range (possibly <code>null</code>).
23 7		*
23 8		* @since 1.0.15
23 9		*/
24 0		public static Range combineIgnoringNaN(Range range1, Range range2)
24 1	<u>4</u>	{ if (range1 == null) {
24 2	<u>1</u> <u>3</u>	if (range2 != null && range2.isNaNRange()) {
24 3	<u>1</u>	return null;
24 4		}
24 5	<u>2</u>	return range2;
24 6		}
24 7	<u>4</u>	if (range2 == null) {
24 8	<u>9</u>	if (range1.isNaNRange()) {
24 9	<u>1</u>	return null;
25 0		}

25 1	<u>2</u>	return range1;
25 2		}
25 3	<u>4</u>	double l = min(range1.getLowerBound(), range2.getLowerBound());
25 4	<u>4</u>	double u = max(range1.getUpperBound(), range2.getUpperBound());
25 5	<u>2</u> <u>8</u>	if (Double.isNaN(l) && Double.isNaN(u)) {
25 6	<u>1</u>	return null;
25 7		}
25 8	<u>1</u> <u>3</u>	return new Range(l, u);
25 9		}
26 0		
26 1		/**
26 2	is	* Returns the minimum value. If either value is NaN, the other value
26 3		* returned. If both are NaN, NaN is returned.
26 4		*
26 5		* @param d1 value 1.
26 6		* @param d2 value 2.
26 7		*

26 8		* @return The minimum of the two values.
26 9		*/
27 0		private static double min(double d1, double d2) {
27 1	<u>1</u> <u>4</u>	if (Double.isNaN(d1)) {
27 2	<u>7</u>	return d2;
27 3		}
27 4	<u>1</u> <u>4</u>	if (Double.isNaN(d2)) {
27 5	<u>7</u>	return d1;
27 6		}
27 7	<u>1</u> <u>4</u>	return Math.min(d1, d2);
27 8		}
27 9		
28 0		private static double max(double d1, double d2) {
28 1	<u>1</u> <u>4</u>	if (Double.isNaN(d1)) {
28 2	<u>7</u>	return d2;
28 3		}
28 4	<u>1</u> <u>4</u>	if (Double.isNaN(d2)) {

28 5	<u>7</u>	return d1;
28 6		}
28 7	<u>1</u> <u>4</u>	return Math.max(d1, d2);
28 8		}
28 9		
29 0		/**
29 1		* Returns a range that includes all the values in the specified
29 2		* <code>range</code> AND the specified <code>value</code> .
29 3		*
29 4		* @param range the range (<code>null</code> permitted).
29 5		* @param value the value that must be included.
29 6		*
29 7		* @return A range.
29 8		*
29 9		* @since 1.0.1
30 0		*/
30 1		public static Range expandToInclude(Range range, double value) {

30 2	<u>4</u>	if (range == null) {
30 3	<u>1</u> <u>3</u>	return new Range(value, value);
30 4		}
30 5	<u>1</u> <u>5</u>	if (value < range.getLowerBound()) {
30 6	<u>9</u>	return new Range(value, range.getUpperBound());
30 7		}
30 8	<u>1</u> <u>5</u>	else if (value > range.getUpperBound()) {
30 9	<u>9</u>	return new Range(range.getLowerBound(), value);
31 0		}
31 1		else {
31 2	<u>2</u>	return range;
31 3		}
31 4		}
31 5		
31 6		/**
31 7		* Creates a new range by adding margins to an existing range.
31 8		*

31 9		* @param range the range (<code>>null</code> not permitted).
32 0		* @param lowerMargin the lower margin (expressed as a percentage of the
32 1		* range length).
32 2		* @param upperMargin the upper margin (expressed as a percentage of the
32 3		* range length).
32 4		*
32 5		* @return The expanded range.
32 6		*/
32 7		public static Range expand(Range range,
32 8		double lowerMargin, double upperMargin) {
32 9	<u>1</u>	ParamChecks.nullNotPermitted(range, "range");
33 0	<u>1</u>	double length = range.getLength();
33 1	<u>2</u> <u>5</u>	double lower = range.getLowerBound() - length * lowerMargin;
33 2	<u>2</u> <u>5</u>	double upper = range.getUpperBound() + length * upperMargin;
33 3	<u>1</u> <u>9</u>	if (lower > upper) {
33 4	<u>4</u> <u>5</u>	lower = lower / 2.0 + upper / 2.0;
33 5	<u>5</u>	upper = lower;

33 6		}
33 7	<u>1</u> <u>3</u>	return new Range(lower, upper);
33 8		}
33 9		
34 0		/**
34 1		* Shifts the range by the specified amount.
34 2		*
34 3		* @param base the base range (<code>>null</code> not permitted).
34 4		* @param delta the shift amount.
34 5		*
34 6		* @return A new range.
34 7		*/
34 8		public static Range shift(Range base, double delta) {
34 9	<u>1</u> <u>4</u>	return shift(base, delta, false);
35 0		}
35 1		
35 2		/**

35	3		* Shifts the range by the specified amount.
35	4		*
35	5		* @param base the base range (<code>null</code> not permitted).
35	6		* @param delta the shift amount.
35	7		* @param allowZeroCrossing a flag that determines whether or not the
35	8		cross
35	9		zero after adjustment.
36	0		*
36	1		* @return A new range.
36	2		*/
36	3		public static Range shift(Range base, double delta,
36	4		boolean allowZeroCrossing) {
36	5	<u>1</u>	ParamChecks.nullNotPermitted(base, "base");
36	6	<u>1</u> <u>3</u>	if (allowZeroCrossing) {
36	7	<u>1</u> <u>6</u>	return new Range(base.getLowerBound() + delta,
36	8	<u>1</u> <u>3</u>	base.getUpperBound() + delta);
36	9		}

37 0		else {
37 1	<u>6</u>	return new
		Range(shiftWithNoZeroCrossing(base.getLowerBound(),
37 2	<u>8</u>	delta),
		shiftWithNoZeroCrossing(base.getUpperBound(),
37 3	<u>5</u>	delta));
37 4		}
37 5		}
37 6		
37 7		/**
37 8		* Returns the given <code>value</code> adjusted by
		<code>delta</code> but
37 9		* with a check to prevent the result from crossing <code>0.0</code> .
38 0		*
38 1		* @param value the value.
38 2		* @param delta the adjustment.
38 3		*
38 4		* @return The adjusted value.
38 5		*/
38 6		private static double shiftWithNoZeroCrossing(double value, double
		delta) {

38 7	<u>1</u> <u>9</u>	if (value > 0.0) {
38 8	<u>2</u> <u>6</u>	return Math.max(value + delta, 0.0);
38 9		}
39 0	<u>1</u> <u>9</u>	else if (value < 0.0) {
39 1	<u>2</u> <u>6</u>	return Math.min(value + delta, 0.0);
39 2		}
39 3		else {
39 4	<u>1</u> <u>9</u>	return value + delta;
39 5		}
39 6		}
39 7		
39 8		/**
39 9		* Scales the range by the specified factor.
40 0		*
40 1		* @param base the base range (<code>>null</code> not permitted).
40 2		* @param factor the scaling factor (must be non-negative).
40 3		*

40			* @return A new range.
4			
40			*
5			
40			* @since 1.0.9
6			
40			*/
7			
40			public static Range scale(Range base, double factor) {
8			
40	<u>1</u>		ParamChecks.nullNotPermitted(base, "base");
9			
41	<u>1</u>		if (factor < 0) {
0	<u>9</u>		
41	<u>1</u>		throw new IllegalArgumentException("Negative 'factor' argument.");
1			
41			}
2			
41	<u>1</u>		return new Range(base.getLowerBound() * factor,
3	<u>6</u>		
41	<u>1</u>		base.getUpperBound() * factor);
4	<u>3</u>		
41			}
5			
41			
6			
41			/**
7			
41			* Tests this object for equality with an arbitrary object.
8			
41			*
9			
42			* @param obj the object to test against (<code>>null</code>
0			permitted).

42	1	*
42	2	* @return A boolean.
42	3	*/
42	4	@Override
42	5	public boolean equals(Object obj) {
42	6	<u>8</u> if (!(obj instanceof Range)) {
42	7	<u>7</u> return false;
42	8	}
42	9	Range range = (Range) obj;
43	0	<u>1</u> <u>8</u> if (!(this.lower == range.lower)) {
43	1	<u>7</u> return false;
43	2	}
43	3	<u>1</u> <u>8</u> if (!(this.upper == range.upper)) {
43	4	<u>7</u> return false;
43	5	}
43	6	<u>8</u> return true;
43	7	}

43		
8		
43		
9		/**
44		
0		* Returns <code>true</code> if both the lower and upper bounds are
44		
1		* <code>Double.NaN</code> , and <code>false</code> otherwise.
44		
2		*
44		
3		* @return A boolean.
44		
4		*
44		
5		* @since 1.0.18
44		
6		*/
44		
7		public boolean isNaNRange() {
44	<u>4</u>	
8	<u>3</u>	return Double.isNaN(this.lower) && Double.isNaN(this.upper);
44		
9		}
45		
0		
45		
1		/**
45		
2		* Returns a hash code.
45		
3		*
45		
4		* @return A hash code.

45		*/
5		
45		@Override
6		
45		public int hashCode() {
7		
45		int result;
8		
45		long temp;
9		
46	<u>6</u>	temp = Double.doubleToLongBits(this.lower);
0		
46	<u>1</u> <u>9</u>	result = (int) (temp ^ (temp >>> 32));
1		
46	<u>6</u>	temp = Double.doubleToLongBits(this.upper);
2		
46	<u>4</u> <u>5</u>	result = 29 * result + (int) (temp ^ (temp >>> 32));
3		
46	<u>7</u>	return result;
4		
46		}
5		
46		
6		
46		/**
7		
46		* Returns a string representation of this Range.
8		
46		*
9		
47		* @return A String "Range[lower,upper]" where lower=lower range
0	and	
47		* upper=upper range.
1		

47	2	*/
47	3	@Override
47	4	public String toString() {
47	5	return ("Range[" + this.lower + "," + this.upper + "]);
47	6	}
47	7	
47	8	}

Mutations

[90](#)

1. changed conditional boundary → KILLED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → SURVIVED
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
10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Incremented (a++) double local variable number 3 → KILLED
14. Decrementd (a--) double local variable number 1 → KILLED
15. Decrementd (a--) double local variable number 3 → KILLED
16. Incremented (++a) double local variable number 1 → KILLED
17. Incremented (++a) double local variable number 3 → KILLED
18. Decrementd (--a) double local variable number 1 → KILLED
19. Decrementd (--a) double local variable number 3 → KILLED

[91](#)

1. removed call to java/lang/StringBuilder::<init> → NO_COVERAGE
2. removed call to java/lang/StringBuilder::append → NO_COVERAGE

3. removed call to java/lang/StringBuilder::toString → NO_COVERAGE
4. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
5. Negated double local variable number 1 → NO_COVERAGE
6. Incremented (a++) double local variable number 1 → NO_COVERAGE
7. Decrementd (a--) double local variable number 1 → NO_COVERAGE
8. Incremented (++a) double local variable number 1 → NO_COVERAGE
9. Decrementd (--a) double local variable number 1 → NO_COVERAGE

92

1. removed call to java/lang/StringBuilder::append → NO_COVERAGE
2. removed call to java/lang/StringBuilder::append → NO_COVERAGE
3. removed call to java/lang/StringBuilder::append → NO_COVERAGE
4. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
5. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
6. replaced call to java/lang/StringBuilder::append with receiver → NO_COVERAGE
7. Negated double local variable number 3 → NO_COVERAGE
8. Incremented (a++) double local variable number 3 → NO_COVERAGE
9. Decrementd (a--) double local variable number 3 → NO_COVERAGE
10. Incremented (++a) double local variable number 3 → NO_COVERAGE
11. Decrementd (--a) double local variable number 3 → NO_COVERAGE

93

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

95

1. Removed assignment to member variable lower → KILLED
2. Negated double local variable number 1 → KILLED
3. Incremented (a++) double local variable number 1 → SURVIVED
4. Decrementd (a--) double local variable number 1 → SURVIVED
5. Incremented (++a) double local variable number 1 → KILLED
6. Decrementd (--a) double local variable number 1 → KILLED

96

1. Removed assignment to member variable upper → KILLED
2. Negated double local variable number 3 → KILLED
3. Incremented (a++) double local variable number 3 → SURVIVED
4. Decrementd (a--) double local variable number 3 → SURVIVED
5. Incremented (++a) double local variable number 3 → KILLED
6. Decrementd (--a) double local variable number 3 → KILLED

10
5

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

11
4

3. Negated double field lower → KILLED
4. Incremented (a++) double field lower → KILLED
5. Decrementd (a--) double field lower → KILLED
6. Incremented (++a) double field lower → KILLED
7. Decrementd (--a) double fieldlower → KILLED

1. replaced double return with 0.0d for
org/jfree/data/Range::getUpperBound → KILLED
2. replaced return of double value with -(x + 1) for
org/jfree/data/Range::getUpperBound → KILLED
3. Negated double field upper → KILLED
4. Incremented (a++) double field upper → KILLED
5. Decrementd (a--) double field upper → KILLED
6. Incremented (++a) double field upper → KILLED
7. Decrementd (--a) double fieldupper → KILLED

12
3

1. Replaced double subtraction with addition → KILLED
2. replaced double return with 0.0d for org/jfree/data/Range::getLength → KILLED
3. replaced return of double value with -(x + 1) for
org/jfree/data/Range::getLength → KILLED
4. Negated double field upper → KILLED
5. Negated double field lower → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation by second member → KILLED
8. Replaced double subtraction with addition → KILLED
9. Replaced double subtraction with multiplication → KILLED
10. Replaced double subtraction with division → KILLED
11. Replaced double subtraction with modulus → KILLED
12. Incremented (a++) double field upper → KILLED
13. Incremented (a++) double field lower → KILLED
14. Decrementd (a--) double field upper → KILLED
15. Decrementd (a--) double field lower → KILLED
16. Incremented (++a) double field upper → KILLED
17. Incremented (++a) double field lower → KILLED
18. Decrementd (--a) double fieldupper → KILLED
19. Decrementd (--a) double fieldlower → KILLED

13
2

1. Substituted 2.0 with 1.0 → KILLED
2. Substituted 2.0 with 1.0 → KILLED
3. Replaced double division with multiplication → KILLED
4. Replaced double division with multiplication → KILLED
5. Replaced double addition with subtraction → KILLED
6. replaced double return with 0.0d for
org/jfree/data/Range::getCentralValue → KILLED

7. replaced return of double value with $-(x + 1)$ for `org/jfree/data/Range::getCentralValue` → KILLED
8. Negated double field lower → KILLED
9. Negated double field upper → KILLED
10. Replaced double operation with first member → KILLED
11. Replaced double operation with first member → KILLED
12. Replaced double operation with first member → KILLED
13. Replaced double operation by second member → KILLED
14. Replaced double operation by second member → KILLED
15. Replaced double operation by second member → KILLED
16. Replaced double division with multiplication → KILLED
17. Replaced double division with multiplication → KILLED
18. Replaced double addition with subtraction → KILLED
19. Replaced double division with modulus → KILLED
20. Replaced double division with modulus → KILLED
21. Replaced double addition with multiplication → KILLED
22. Replaced double division with addition → KILLED
23. Replaced double division with addition → KILLED
24. Replaced double addition with division → KILLED
25. Replaced double division with subtraction → KILLED
26. Replaced double division with subtraction → KILLED
27. Replaced double addition with modulus → KILLED
28. Substituted 2.0 with 1.0 → KILLED
29. Substituted 2.0 with 1.0 → KILLED
30. Substituted 2.0 with 0.0 → KILLED
31. Substituted 2.0 with 0.0 → KILLED
32. Substituted 2.0 with -1.0 → KILLED
33. Substituted 2.0 with -1.0 → KILLED
34. Substituted 2.0 with -2.0 → KILLED
35. Substituted 2.0 with -2.0 → KILLED
36. Substituted 2.0 with 3.0 → KILLED
37. Substituted 2.0 with 3.0 → KILLED
38. Substituted 2.0 with 1.0 → KILLED
39. Substituted 2.0 with 1.0 → KILLED
40. Incremented (a++) double field lower → SURVIVED
41. Incremented (a++) double field upper → SURVIVED
42. Decrementd (a--) double field lower → SURVIVED
43. Decrementd (a--) double field upper → SURVIVED
44. Incremented (++a) double field lower → KILLED
45. Incremented (++a) double field upper → KILLED
46. Decrementd (--a) double fieldlower → KILLED
47. Decrementd (--a) double fieldupper → KILLED

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 → 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 → 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 → SURVIVED
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 → SURVIVED
39. Incremented (a++) double field lower → KILLED
40. Incremented (a++) double local variable number 1 → SURVIVED
41. Incremented (a++) double field upper → KILLED
42. Decrementd (a--) double local variable number 1 → KILLED

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

1. changed conditional boundary → 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 than to less than → KILLED
 8. greater than to less or equal → KILLED
 9. greater than to greater or equal → SURVIVED
 10. greater than to equal → KILLED
 11. greater than to not equal → KILLED
 12. Incremented (a++) double local variable number 1 → SURVIVED
 13. Incremented (a++) double field lower → SURVIVED
 14. Decrementd (a--) double local variable number 1 → SURVIVED
 15. Decrementd (a--) double field lower → KILLED
 16. Incremented (++a) double local variable number 1 → SURVIVED
 17. Incremented (++a) double field lower → SURVIVED
 18. Decrementd (--a) double local variable number 1 → SURVIVED
 19. Decrementd (--a) double fieldlower → KILLED

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED
 2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED
 3. changed conditional boundary → SURVIVED
 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

15
7

15
8

11. Negated double local variable number 3 → KILLED
12. Negated double field lower → KILLED
13. Substituted 0 with 1 → KILLED
14. Substituted 1 with 0 → KILLED
15. Substituted 1 with -1 → SURVIVED
16. Substituted 0 with -1 → KILLED
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 → SURVIVED
23. Less or equal to greater than → KILLED
24. Less or equal to greater or equal → KILLED
25. Less or equal to equal → KILLED
26. Less or equal to not equal → KILLED
27. Incremented (a++) double local variable number 3 → SURVIVED
28. Incremented (a++) double field lower → SURVIVED
29. Decrementd (a--) double local variable number 3 → SURVIVED
30. Decrementd (a--) double field lower → SURVIVED
31. Incremented (++a) double local variable number 3 → KILLED
32. Incremented (++a) double field lower → SURVIVED
33. Decrementd (--a) double local variable number 3 → SURVIVED
34. Decrementd (--a) double fieldlower → KILLED

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED
2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED
3. changed conditional boundary → SURVIVED
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 → SURVIVED
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 → SURVIVED
39. Incremented (a++) double field upper → SURVIVED
40. Incremented (a++) double local variable number 3 → SURVIVED
41. Incremented (a++) double local variable number 1 → SURVIVED
42. Decrementd (a--) double local variable number 1 → SURVIVED
43. Decrementd (a--) double field upper → SURVIVED
44. Decrementd (a--) double local variable number 3 → SURVIVED
45. Decrementd (a--) double local variable number 1 → SURVIVED
46. Incremented (++a) double local variable number 1 → SURVIVED
47. Incremented (++a) double field upper → SURVIVED
48. Incremented (++a) double local variable number 3 → SURVIVED
49. Incremented (++a) double local variable number 1 → SURVIVED
50. Decrementd (--a) double local variable number 1 → SURVIVED
51. Decrementd (--a) double fieldupper → SURVIVED
52. Decrementd (--a) double local variable number 3 → SURVIVED
53. Decrementd (--a) double local variable number 1 → SURVIVED

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED
2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → KILLED
4. removed call to org/jfree/data/Range::getUpperBound → KILLED

18
8

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

- 1. Negated double local variable number 1 → KILLED
- 2. Incremented (a++) double local variable number 1 → SURVIVED
- 3. Decrementd (a--) double local variable number 1 → SURVIVED
- 4. Incremented (++a) double local variable number 1 → KILLED
- 5. Decrementd (--a) double local variable number 1 → KILLED

18
9

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

19
0

- 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 → SURVIVED
- 7. Less or equal to less than → SURVIVED
- 8. Less or equal to greater than → KILLED
- 9. Less or equal to greater or equal → KILLED
- 10. Less or equal to equal → KILLED
- 11. Less or equal to not equal → KILLED
- 12. Incremented (a++) double local variable number 1 → SURVIVED
- 13. Incremented (a++) double field upper → KILLED
- 14. Decrementd (a--) double local variable number 1 → SURVIVED
- 15. Decrementd (a--) double field upper → KILLED
- 16. Incremented (++a) double local variable number 1 → SURVIVED
- 17. Incremented (++a) double field upper → KILLED
- 18. Decrementd (--a) double local variable number 1 → SURVIVED
- 19. Decrementd (--a) double fieldupper → KILLED

19
1

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

19
3

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

19
4

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

19
7

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

21
7

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
<u>21</u> <u>8</u>	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
<u>22</u> <u>0</u>	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
<u>22</u> <u>1</u>	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
<u>22</u> <u>3</u>	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 → SURVIVED 4. removed call to java/lang/Math::min → KILLED
<u>22</u> <u>4</u>	1. replaced call to java/lang/Math::max with argument → SURVIVED 2. removed call to org/jfree/data/Range::getUpperBound → SURVIVED 3. removed call to org/jfree/data/Range::getUpperBound → KILLED 4. removed call to java/lang/Math::max → KILLED
<u>22</u> <u>5</u>	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. Decrementd (a--) double local variable number 2 → SURVIVED 9. Decrementd (a--) double local variable number 4 → SURVIVED 10. Incremented (++a) double local variable number 2 → KILLED 11. Incremented (++a) double local variable number 4 → KILLED 12. Decrementd (--a) double local variable number 2 → KILLED 13. Decrementd (--a) double local variable number 4 → KILLED
<u>24</u> <u>1</u>	1. negated conditional → KILLED 2. removed conditional - replaced equality check with false → KILLED 3. removed conditional - replaced equality check with true → KILLED

24
2

4. not equal to equal → KILLED

1. negated conditional → KILLED
2. negated conditional → KILLED
3. removed call to org/jfree/data/Range::isNaNRange → KILLED
4. removed conditional - replaced equality check with false → KILLED
5. removed conditional - replaced equality check with false → KILLED
6. removed conditional - replaced equality check with true → KILLED
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

24
3

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

24
5

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

24
7

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

24
8

1. negated conditional → KILLED
2. removed call to org/jfree/data/Range::isNaNRange → KILLED
3. removed conditional - replaced equality check with false → KILLED
4. removed conditional - replaced equality check with true → KILLED
5. equal to less than → KILLED
6. equal to less or equal → SURVIVED
7. equal to greater than → KILLED
8. equal to greater or equal → KILLED
9. equal to not equal → KILLED

24
9

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

25
1

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

25
3

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

25
4

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

25
5

1. negated conditional → KILLED
2. negated conditional → KILLED
3. removed call to java/lang/Double::isNaN → KILLED
4. removed call to java/lang/Double::isNaN → KILLED
5. removed conditional - replaced equality check with false → KILLED
6. removed conditional - replaced equality check with false → KILLED
7. removed conditional - replaced equality check with true → KILLED
8. removed conditional - replaced equality check with true → KILLED
9. Negated double local variable number 2 → SURVIVED
10. Negated double local variable number 4 → SURVIVED
11. equal to less than → KILLED
12. equal to less than → KILLED
13. equal to less or equal → SURVIVED
14. equal to less or equal → SURVIVED
15. equal to greater than → KILLED
16. equal to greater than → KILLED
17. equal to greater or equal → KILLED
18. equal to greater or equal → KILLED
19. equal to not equal → KILLED
20. equal to not equal → KILLED
21. Incremented (a++) double local variable number 2 → KILLED
22. Incremented (a++) double local variable number 4 → SURVIVED
23. Decrementd (a--) double local variable number 2 → KILLED
24. Decrementd (a--) double local variable number 4 → SURVIVED
25. Incremented (++a) double local variable number 2 → KILLED
26. Incremented (++a) double local variable number 4 → SURVIVED
27. Decrementd (--a) double local variable number 2 → KILLED
28. Decrementd (--a) double local variable number 4 → SURVIVED

25
6

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

25
8

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

27
1

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

27
2

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

27
4

1. negated conditional → SURVIVED
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 → SURVIVED
5. Negated double local variable number 2 → SURVIVED
6. equal to less than → SURVIVED
7. equal to less or equal → SURVIVED
8. equal to greater than → SURVIVED
9. equal to greater or equal → SURVIVED
10. equal to not equal → SURVIVED
11. Incremented (a++) double local variable number 2 → SURVIVED
12. Decrementd (a--) double local variable number 2 → SURVIVED
13. Incremented (++a) double local variable number 2 → SURVIVED
14. Decrementd (--a) double local variable number 2 → SURVIVED

27
5

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

27
7

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

28
1

1. negated conditional → SURVIVED
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 → SURVIVED
5. Negated double local variable number 0 → SURVIVED
6. equal to less than → SURVIVED
7. equal to less or equal → SURVIVED
8. equal to greater than → SURVIVED
9. equal to greater or equal → SURVIVED
10. equal to not equal → SURVIVED
11. Incremented (a++) double local variable number 0 → SURVIVED
12. Decrementd (a--) double local variable number 0 → SURVIVED
13. Incremented (++a) double local variable number 0 → SURVIVED
14. Decrementd (--a) double local variable number 0 → SURVIVED

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 → SURVIVED
4. Incremented (a++) double local variable number 2 → SURVIVED
5. Decrementd (a--) double local variable number 2 → SURVIVED
6. Incremented (++a) double local variable number 2 → SURVIVED
7. Decrementd (--a) double local variable number 2 → SURVIVED

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

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

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

3. Negated double local variable number 0 → NO_COVERAGE
4. Incremented (a++) double local variable number 0 → NO_COVERAGE
5. Decrementd (a--) double local variable number 0 → NO_COVERAGE
6. Incremented (++a) double local variable number 0 → NO_COVERAGE

	7. Decrementd (--a) double local variable number 0 → NO_COVERAGE
	1. replaced call to java/lang/Math::max with argument → SURVIVED
	2. removed call to java/lang/Math::max → KILLED
	3. replaced double return with 0.0d for org/jfree/data/Range::max → KILLED
	4. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → KILLED
	5. Negated double local variable number 0 → SURVIVED
28	6. Negated double local variable number 2 → KILLED
7	7. Incremented (a++) double local variable number 0 → SURVIVED
	8. Incremented (a++) double local variable number 2 → SURVIVED
	9. Decrementd (a--) double local variable number 0 → SURVIVED
	10. Decrementd (a--) double local variable number 2 → SURVIVED
	11. Incremented (++a) double local variable number 0 → SURVIVED
	12. Incremented (++a) double local variable number 2 → KILLED
	13. Decrementd (--a) double local variable number 0 → SURVIVED
	14. Decrementd (--a) double local variable number 2 → KILLED
	1. negated conditional → KILLED
30	2. removed conditional - replaced equality check with false → KILLED
2	3. removed conditional - replaced equality check with true → KILLED
	4. not equal to equal → KILLED
	1. removed call to org/jfree/data/Range::<init> → KILLED
	2. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED
	3. mutated return of Object value for org/jfree/data/Range::expandToInclude to (if (x != null) null else throw new RuntimeException) → KILLED
	4. Negated double local variable number 1 → SURVIVED
30	5. Negated double local variable number 1 → SURVIVED
3	6. Incremented (a++) double local variable number 1 → KILLED
	7. Incremented (a++) double local variable number 1 → SURVIVED
	8. Decrementd (a--) double local variable number 1 → KILLED
	9. Decrementd (a--) double local variable number 1 → SURVIVED
	10. Incremented (++a) double local variable number 1 → KILLED
	11. Incremented (++a) double local variable number 1 → KILLED
	12. Decrementd (--a) double local variable number 1 → KILLED
	13. Decrementd (--a) double local variable number 1 → KILLED
	1. changed conditional boundary → SURVIVED
30	2. negated conditional → KILLED
5	3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
	4. removed conditional - replaced comparison check with false → KILLED

5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → KILLED
7. greater or equal to less than → KILLED
8. greater or equal to less or equal → KILLED
9. greater or equal to greater than → SURVIVED
10. greater or equal to equal → KILLED
11. greater or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Decrementd (a--) double local variable number 1 → KILLED
14. Incremented (++a) double local variable number 1 → KILLED
15. Decrementd (--a) double local variable number 1 → KILLED

30
6

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

30
8

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 → KILLED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → KILLED
7. Less or equal to less than → SURVIVED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Decrementd (a--) double local variable number 1 → KILLED
14. Incremented (++a) double local variable number 1 → KILLED
15. Decrementd (--a) double local variable number 1 → KILLED

30
9

1. removed call to org/jfree/data/Range::<init> → KILLED
2. removed call to org/jfree/data/Range::getLowerBound → KILLED
3. replaced return value with null for
org/jfree/data/Range::expandToInclude → KILLED

4. mutated return of Object value for
org/jfree/data/Range::expandToInclude to (if (x != null) null else throw
new RuntimeException) → KILLED
5. Negated double local variable number 1 → KILLED
6. Incremented (a++) double local variable number 1 → SURVIVED
7. Decrementd (a--) double local variable number 1 → SURVIVED
8. Incremented (++a) double local variable number 1 → KILLED
9. Decrementd (--a) double local variable number 1 → KILLED

31
2

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

32
9

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

33
0

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

33
1

1. Replaced double multiplication with division → KILLED
2. Replaced double subtraction with addition → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → KILLED
4. Negated double local variable number 5 → KILLED
5. Negated double local variable number 1 → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → KILLED
9. Replaced double operation by second member → KILLED
10. Replaced double multiplication with division → KILLED
11. Replaced double subtraction with addition → KILLED
12. Replaced double multiplication with modulus → KILLED
13. Replaced double subtraction with multiplication → KILLED
14. Replaced double multiplication with addition → KILLED
15. Replaced double subtraction with division → KILLED
16. Replaced double multiplication with subtraction → KILLED
17. Replaced double subtraction with modulus → KILLED
18. Incremented (a++) double local variable number 5 → KILLED
19. Incremented (a++) double local variable number 1 → SURVIVED
20. Decrementd (a--) double local variable number 5 → KILLED
21. Decrementd (a--) double local variable number 1 → SURVIVED
22. Incremented (++a) double local variable number 5 → KILLED
23. Incremented (++a) double local variable number 1 → KILLED
24. Decrementd (--a) double local variable number 5 → KILLED

33
2

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

1. Replaced double multiplication with division → KILLED
2. Replaced double addition with subtraction → KILLED
3. removed call to org/jfree/data/Range::getUpperBound → KILLED
4. Negated double local variable number 5 → KILLED
5. Negated double local variable number 3 → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → KILLED
9. Replaced double operation by second member → KILLED
10. Replaced double multiplication with division → KILLED
11. Replaced double addition with subtraction → KILLED
12. Replaced double multiplication with modulus → KILLED
13. Replaced double addition with multiplication → KILLED
14. Replaced double multiplication with addition → KILLED
15. Replaced double addition with division → KILLED
16. Replaced double multiplication with subtraction → KILLED
17. Replaced double addition with modulus → KILLED
18. Incremented (a++) double local variable number 5 → SURVIVED
19. Incremented (a++) double local variable number 3 → SURVIVED
20. Decrementd (a--) double local variable number 5 → SURVIVED
21. Decrementd (a--) double local variable number 3 → SURVIVED
22. Incremented (++a) double local variable number 5 → KILLED
23. Incremented (++a) double local variable number 3 → KILLED
24. Decrementd (--a) double local variable number 5 → KILLED
25. Decrementd (--a) double local variable number 3 → KILLED

33
3

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

17. Incremented (++a) double local variable number 9 → KILLED
 18. Decrementd (--a) double local variable number 7 → KILLED
 19. Decrementd (--a) double local variable number 9 → KILLED

1. Substituted 2.0 with 1.0 → KILLED
 2. Substituted 2.0 with 1.0 → KILLED
 3. Replaced double division with multiplication → KILLED
 4. Replaced double division with multiplication → KILLED
 5. Replaced double addition with subtraction → KILLED
 6. Negated double local variable number 7 → KILLED
 7. Negated double local variable number 9 → KILLED
 8. Replaced double operation with first member → KILLED
 9. Replaced double operation with first member → KILLED
 10. Replaced double operation with first member → KILLED
 11. Replaced double operation by second member → KILLED
 12. Replaced double operation by second member → KILLED
 13. Replaced double operation by second member → KILLED
 14. Replaced double division with multiplication → KILLED
 15. Replaced double division with multiplication → KILLED
 16. Replaced double addition with subtraction → KILLED
 17. Replaced double division with modulus → KILLED
 18. Replaced double division with modulus → KILLED
 19. Replaced double addition with multiplication → KILLED
 20. Replaced double division with addition → KILLED
 21. Replaced double division with addition → KILLED
 22. Replaced double addition with division → KILLED
 23. Replaced double division with subtraction → KILLED
 24. Replaced double division with subtraction → KILLED

25. Replaced double addition with modulus → SURVIVED

26. Substituted 2.0 with 1.0 → KILLED
 27. Substituted 2.0 with 1.0 → KILLED
 28. Substituted 2.0 with 0.0 → KILLED
 29. Substituted 2.0 with 0.0 → KILLED
 30. Substituted 2.0 with -1.0 → KILLED
 31. Substituted 2.0 with -1.0 → KILLED
 32. Substituted 2.0 with -2.0 → KILLED
 33. Substituted 2.0 with -2.0 → KILLED
 34. Substituted 2.0 with 3.0 → KILLED
 35. Substituted 2.0 with 3.0 → KILLED
 36. Substituted 2.0 with 1.0 → KILLED
 37. Substituted 2.0 with 1.0 → KILLED

38. Incremented (a++) double local variable number 7 → SURVIVED
 39. Incremented (a++) double local variable number 9 → SURVIVED
 40. Decrementd (a--) double local variable number 7 → SURVIVED

33
5

41. Decrementd (a--) double local variable number 9 → SURVIVED
42. Incremented (a++) double local variable number 7 → KILLED
43. Incremented (a++) double local variable number 9 → KILLED
44. Decrementd (--a) double local variable number 7 → KILLED
45. Decrementd (--a) double local variable number 9 → KILLED

1. Negated double local variable number 7 → SURVIVED
2. Incremented (a++) double local variable number 7 → KILLED
3. Decrementd (a--) double local variable number 7 → KILLED
4. Incremented (a++) double local variable number 7 → KILLED
5. Decrementd (--a) double local variable number 7 → KILLED

33
7

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

34
9

1. replaced call to org/jfree/data/Range::shift with argument → KILLED
2. Substituted 0 with 1 → KILLED
3. removed call to org/jfree/data/Range::shift → KILLED
4. replaced return value with null for org/jfree/data/Range::shift → KILLED
5. mutated return of Object value for org/jfree/data/Range::shift to (if (x != null) null else throw new RuntimeException) → KILLED
6. Negated double local variable number 1 → KILLED
7. Substituted 0 with 1 → KILLED
8. Substituted 0 with -1 → KILLED
9. Substituted 0 with 1 → KILLED
10. Substituted 0 with -1 → KILLED
11. Incremented (a++) double local variable number 1 → SURVIVED
12. Decrementd (a--) double local variable number 1 → SURVIVED
13. Incremented (a++) double local variable number 1 → KILLED
14. Decrementd (--a) double local variable number 1 → KILLED

36
5

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

36
6

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

36
7

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

36
8

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

37
1

11. Decrement (a--) double local variable number 1 → SURVIVED
12. Increment (++) double local variable number 1 → KILLED
13. Decrement (--a) double local variable number 1 → KILLED

1. replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument → KILLED
2. removed call to org/jfree/data/Range::<init> → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → KILLED
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

37
2

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. Increment (a++) double local variable number 1 → KILLED
6. Decrement (a--) double local variable number 1 → KILLED
7. Increment (++) double local variable number 1 → KILLED
8. Decrement (--a) double local variable number 1 → KILLED

37
3

1. Negated double local variable number 1 → KILLED
2. Increment (a++) double local variable number 1 → SURVIVED
3. Decrement (a--) double local variable number 1 → SURVIVED
4. Increment (++) double local variable number 1 → KILLED
5. Decrement (--a) double local variable number 1 → KILLED

38
7

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 → SURVIVED
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 → SURVIVED
 16. Incremented (a++) double local variable number 0 → KILLED
 17. Decrementd (a--) double local variable number 0 → KILLED
 18. Incremented (++a) double local variable number 0 → KILLED
 19. Decrementd (--a) double local variable number 0 → KILLED

1. replaced call to java/lang/Math::max with argument → KILLED
 2. Substituted 0.0 with 1.0 → SURVIVED
 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 → SURVIVED
 16. Substituted 0.0 with -1.0 → SURVIVED
 17. Substituted 0.0 with 1.0 → SURVIVED
 18. Substituted 0.0 with -1.0 → SURVIVED
 19. Incremented (a++) double local variable number 0 → SURVIVED
 20. Incremented (a++) double local variable number 2 → SURVIVED
 21. Decrementd (a--) double local variable number 0 → SURVIVED
 22. Decrementd (a--) double local variable number 2 → SURVIVED
 23. Incremented (++a) double local variable number 0 → KILLED
 24. Incremented (++a) double local variable number 2 → KILLED
 25. Decrementd (--a) double local variable number 0 → KILLED
 26. Decrementd (--a) double local variable number 2 → KILLED

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

38
 8

39
 0

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

1. replaced call to java/lang/Math::min with argument → KILLED
2. Substituted 0.0 with 1.0 → KILLED
3. Replaced double addition with subtraction → KILLED
4. removed call to java/lang/Math::min → KILLED
5. replaced double return with 0.0d for
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
6. replaced return of double value with -(x + 1) for
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
7. Negated double local variable number 0 → KILLED
8. Negated double local variable number 2 → KILLED
9. Replaced double operation with first member → KILLED
10. Replaced double operation by second member → KILLED
11. Replaced double addition with subtraction → KILLED
12. Replaced double addition with multiplication → KILLED
13. Replaced double addition with division → KILLED
14. Replaced double addition with modulus → KILLED
15. Substituted 0.0 with 1.0 → KILLED
16. Substituted 0.0 with -1.0 → KILLED
17. Substituted 0.0 with 1.0 → KILLED
18. Substituted 0.0 with -1.0 → KILLED
19. Incremented (a++) double local variable number 0 → SURVIVED
20. Incremented (a++) double local variable number 2 → SURVIVED
21. Decrementd (a--) double local variable number 0 → SURVIVED
22. Decrementd (a--) double local variable number 2 → SURVIVED
23. Incremented (++a) double local variable number 0 → KILLED
24. Incremented (++a) double local variable number 2 → KILLED
25. Decrementd (--a) double local variable number 0 → KILLED
26. Decrementd (--a) double local variable number 2 → KILLED

1. Replaced double addition with subtraction → KILLED
2. replaced double return with 0.0d for
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED

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

4. Negated double local variable number 0 → SURVIVED

5. Negated double local variable number 2 → KILLED

6. Replaced double operation with first member → KILLED

7. Replaced double operation by second member → SURVIVED

8. Replaced double addition with subtraction → KILLED

9. Replaced double addition with multiplication → KILLED

10. Replaced double addition with division → KILLED

11. Replaced double addition with modulus → KILLED

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

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

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

15. Decrementd (a--) double local variable number 2 → SURVIVED

16. Incremented (++a) double local variable number 0 → KILLED

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

18. Decrementd (--a) double local variable number 0 → KILLED

19. Decrementd (--a) double local variable number 2 → KILLED

40
9

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

1. changed conditional boundary → KILLED

2. Substituted 0.0 with 1.0 → KILLED

3. negated conditional → KILLED

4. removed conditional - replaced comparison check with false →
SURVIVED

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

6. Negated double local variable number 1 → KILLED

7. Substituted 0.0 with 1.0 → KILLED

8. Substituted 0.0 with -1.0 → SURVIVED

41
0

9. Substituted 0.0 with 1.0 → KILLED

10. Substituted 0.0 with -1.0 → SURVIVED

11. greater or equal to less than → KILLED

12. greater or equal to less or equal → KILLED

13. greater or equal to greater than → KILLED

14. greater or equal to equal → KILLED

15. greater or equal to not equal → KILLED

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

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

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

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

41
1

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

41
3

1. removed call to org/jfree/data/Range::<init> → KILLED
2. Replaced double multiplication with division → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → KILLED
4. replaced return value with null for org/jfree/data/Range::scale → KILLED
5. mutated return of Object value for org/jfree/data/Range::scale to (if (x != null) null else throw new RuntimeException) → KILLED
6. Negated double local variable number 1 → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → KILLED
9. Replaced double multiplication with division → KILLED
10. Replaced double multiplication with modulus → KILLED
11. Replaced double multiplication with addition → KILLED
12. Replaced double multiplication with subtraction → KILLED
13. Incremented (a++) double local variable number 1 → KILLED
14. Decrementd (a--) double local variable number 1 → KILLED
15. Incremented (++a) double local variable number 1 → KILLED
16. Decrementd (--a) double local variable number 1 → KILLED

41
4

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

42
6

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

42
7

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. 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

43
0

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

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

43
1

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 → SURVIVED
10. equal to not equal → KILLED
11. Incremented (a++) double field upper → KILLED

43
3

43
4

12. Incremented (a++) double field upper → SURVIVED
13. Decrementd (a--) double field upper → KILLED
14. Decrementd (a--) double field upper → SURVIVED
15. Incremented (++a) double field upper → KILLED
16. Incremented (++a) double field upper → KILLED
17. Decrementd (--a) double fieldupper → KILLED
18. Decrementd (--a) double fieldupper → KILLED

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

43
6

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

44
8

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
 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 → SURVIVED
 37. Incremented (a++) double field upper → SURVIVED
 38. Decrementd (a--) double field lower → SURVIVED
 39. Decrementd (a--) double field upper → SURVIVED
 40. Incremented (++a) double field lower → SURVIVED
 41. Incremented (++a) double field upper → SURVIVED
 42. Decrementd (--a) double fieldlower → SURVIVED
 43. Decrementd (--a) double fieldupper → SURVIVED

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

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 1 → SURVIVED
 5. Negated long local variable number 1 → 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

[46](#)
[0](#)

[46](#)
[1](#)

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. Decrementd (a--) long local variable number 1 → SURVIVED
15. Decrementd (a--) long local variable number 1 → SURVIVED
16. Incremented (++a) long local variable number 1 → SURVIVED
17. Incremented (++a) long local variable number 1 → SURVIVED
18. Decrementd (--a) long local variable number 1 → SURVIVED
19. Decrementd (--a) long local variable number 1 → SURVIVED

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

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 3 → SURVIVED
8. Negated long local variable number 1 → SURVIVED
9. Negated long local variable number 1 → 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 → KILLED
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

46
2

46
3

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. Decrementd (a--) integer local variable number 3 → SURVIVED
38. Decrementd (a--) long local variable number 1 → SURVIVED
39. Decrementd (a--) long local variable number 1 → SURVIVED
40. Incremented (++a) integer local variable number 3 → SURVIVED
41. Incremented (++a) long local variable number 1 → SURVIVED
42. Incremented (++a) long local variable number 1 → SURVIVED
43. Decrementd (--a) integer local variable number 3 → SURVIVED
44. Decrementd (--a) long local variable number 1 → SURVIVED
45. Decrementd (--a) long local variable number 1 → SURVIVED

1. replaced int return with 0 for org/jfree/data/Range::hashCode → KILLED
2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
3. Negated integer local variable number 3 → SURVIVED
4. Incremented (a++) integer local variable number 3 → SURVIVED
5. Decrementd (a--) integer local variable number 3 → SURVIVED
6. Incremented (++a) integer local variable number 3 → SURVIVED
7. Decrementd (--a) integer local variable number 3 → SURVIVED

1. removed call to java/lang/StringBuilder::<init> → KILLED
2. replaced return value with "" for org/jfree/data/Range::toString → KILLED
3. removed call to java/lang/StringBuilder::append → KILLED
4. removed call to java/lang/StringBuilder::append → KILLED
5. removed call to java/lang/StringBuilder::append → KILLED
6. removed call to java/lang/StringBuilder::append → KILLED
7. removed call to java/lang/StringBuilder::toString → KILLED
8. mutated return of Object value for org/jfree/data/Range::toString to (if (x != null) null else throw new RuntimeException) → KILLED
9. replaced call to java/lang/StringBuilder::append with receiver → KILLED
10. replaced call to java/lang/StringBuilder::append with receiver → KILLED
11. replaced call to java/lang/StringBuilder::append with receiver → KILLED

46
4

47
5

12. replaced call to java/lang/StringBuilder::append with receiver → KILLED
13. Negated double field lower → KILLED
14. Negated double field upper → KILLED
15. Incremented (a++) double field lower → SURVIVED
16. Incremented (a++) double field upper → SURVIVED
17. Decrementd (a--) double field lower → SURVIVED
18. Decrementd (a--) double field upper → SURVIVED
19. Incremented (++a) double field lower → KILLED
20. Incremented (++a) double field upper → KILLED
21. Decrementd (--a) double fieldlower → KILLED
22. Decrementd (--a) double fieldupper → KILLED

Active mutators

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

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

Tests examined

- org.jfree.data.test.RangeTest (20 ms)

Report generated by [PIT](#) 1.4.11