

DataUtilities.java

```

1  /* =====
2  * JFreeChart : a free chart library for the Java(tm) platform
3  * =====
4  *
5  * (C) Copyright 2000-2013, 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 * DataUtilities.java
29 * -----
30 * (C) Copyright 2003-2013, by Object Refinery Limited and contributors.
31 *
32 * Original Author: David Gilbert (for Object Refinery Limited);
33 * Contributor(s): Peter Kolb (patch 2511330);
34 *
35 * Changes
36 * -----
37 * 05-Mar-2003 : Version 1 (DG);
38 * 03-Mar-2005 : Moved createNumberArray() and createNumberArray2D() methods
39 *               from the DatasetUtilities class (DG);
40 * 17-May-2005 : Added calculateColumnTotal() and calculateRowTotal()
41 *               methods (DG);
42 * 28-Jan-2009 : Added equal(double[][], double[][]) method (DG);
43 * 28-Jan-2009 : Added clone(double[][]) method (DG);
44 * 04-Feb-2009 : Added calculateColumnTotal/RowTotal variants (PK);
45 * 03-Jul-2013 : Use ParamChecks (DG);
46 *
47 */
48
49 package org.jfree.data;
50
51 import java.util.Arrays;
52 import org.jfree.chart.util.ParamChecks;
53 import org.jfree.data.general.DatasetUtilities;
54
55 /**
56 * Utility methods for use with some of the data classes (but not the datasets,
57 * see {@link DatasetUtilities}).
58 */
59 public abstract class DataUtilities {
60
61     /**

```

```

62     * Tests two arrays for equality. To be considered equal, the arrays must
63     * have exactly the same dimensions, and the values in each array must also
64     * match (two values that are both NaN or both INF are considered equal
65     * in this test).
66     *
67     * @param a the first array (<code>null</code> permitted).
68     * @param b the second array (<code>null</code> permitted).
69     *
70     * @return A boolean.
71     *
72     * @since 1.0.13
73     */
74     public static boolean equal(double[][] a, double[][] b) {
75         3         if (a == null) {
76             2             return (b == null);
77         }
78         3         if (b == null) {
79             2             return false; // already know 'a' isn't null
80         }
81         3         if (a.length != b.length) {
82             2             return false;
83         }
84         2         for (int i = 0; i < a.length; i++) {
85             4             if (!Arrays.equals(a[i], b[i])) {
86                 2                 return false;
87             }
88         }
89         2         return true;
90     }
91
92     /**
93     * Returns a clone of the specified array.
94     *
95     * @param source the source array (<code>null</code> not permitted).
96     *
97     * @return A clone of the array.
98     *
99     * @since 1.0.13
100    */
101    public static double[][] clone(double[][] source) {
102        1        ParamChecks.nullNotPermitted(source, "source");
103        double[][] clone = new double[source.length][];
104        2        for (int i = 0; i < source.length; i++) {
105            3            if (source[i] != null) {
106                double[] row = new double[source[i].length];
107            3            System.arraycopy(source[i], 0, row, 0, source[i].length);
108                clone[i] = row;
109            }
110        }
111        1        return clone;
112    }
113
114    /**
115    * Returns the total of the values in one column of the supplied data
116    * table.
117    *
118    * @param data the table of values (<code>null</code> not permitted).
119    * @param column the column index (zero-based).
120    *
121    * @return The total of the values in the specified column.
122    */
123    public static double calculateColumnTotal(Values2D data, int column) {
124        1        ParamChecks.nullNotPermitted(data, "data");
125        1        double total = 0.0;

```

```
126 1      int rowCount = data.getRowCount();
127 2      for (int r = 0; r < rowCount; r++) {
128 1          Number n = data.getValue(r, column);
129 3          if (n != null) {
130 2              total += n.doubleValue();
131          }
132      }
133 1      return total;
134  }
135
136  /**
137   * Returns the total of the values in one column of the supplied data
138   * table by taking only the row numbers in the array into account.
139   *
140   * @param data the table of values (<code>null</code> not permitted).
141   * @param column the column index (zero-based).
142   * @param validRows the array with valid rows (zero-based).
143   *
144   * @return The total of the valid values in the specified column.
145   *
146   * @since 1.0.13
147   */
148  public static double calculateColumnTotal(Values2D data, int column,
149      int[] validRows) {
150  1      ParamChecks.nullNotPermitted(data, "data");
151  1      double total = 0.0;
152  1      int rowCount = data.getRowCount();
153  2      for (int v = 0; v < validRows.length; v++) {
154          int row = validRows[v];
155  4          if (row < rowCount) {
156  1              Number n = data.getValue(row, column);
157  3              if (n != null) {
158  2                  total += n.doubleValue();
159              }
160          }
161      }
162  1      return total;
163  }
164
165  /**
166   * Returns the total of the values in one row of the supplied data
167   * table.
168   *
169   * @param data the table of values (<code>null</code> not permitted).
170   * @param row the row index (zero-based).
171   *
172   * @return The total of the values in the specified row.
173   */
174  public static double calculateRowTotal(Values2D data, int row) {
175  1      ParamChecks.nullNotPermitted(data, "data");
176  1      double total = 0.0;
177  1      int columnCount = data.getColumnCount();
178  2      for (int c = 0; c < columnCount; c++) {
179  1          Number n = data.getValue(row, c);
180  3          if (n != null) {
181  2              total += n.doubleValue();
182          }
183      }
184  1      return total;
185  }
186
187  /**
188   * Returns the total of the values in one row of the supplied data
```

```
189     * table by taking only the column numbers in the array into account.
190     *
191     * @param data the table of values (<code>null</code> not permitted).
192     * @param row the row index (zero-based).
193     * @param validCols the array with valid cols (zero-based).
194     *
195     * @return The total of the valid values in the specified row.
196     *
197     * @since 1.0.13
198     */
199     public static double calculateRowTotal(Values2D data, int row,
200         int[] validCols) {
201         ParamChecks.nullNotPermitted(data, "data");
202         double total = 0.0;
203         int colCount = data.getColumnCount();
204         for (int v = 0; v < validCols.length; v++) {
205             int col = validCols[v];
206             if (col < colCount) {
207                 Number n = data.getValue(row, col);
208                 if (n != null) {
209                     total += n.doubleValue();
210                 }
211             }
212         }
213         return total;
214     }
215
216     /**
217     * Constructs an array of <code>Number</code> objects from an array of
218     * <code>double</code> primitives.
219     *
220     * @param data the data (<code>null</code> not permitted).
221     *
222     * @return An array of <code>Double</code>.
223     */
224     public static Number[] createNumberArray(double[] data) {
225         ParamChecks.nullNotPermitted(data, "data");
226         Number[] result = new Number[data.length];
227         for (int i = 0; i < data.length; i++) {
228             result[i] = new Double(data[i]);
229         }
230         return result;
231     }
232
233     /**
234     * Constructs an array of arrays of <code>Number</code> objects from a
235     * corresponding structure containing <code>double</code> primitives.
236     *
237     * @param data the data (<code>null</code> not permitted).
238     *
239     * @return An array of <code>Double</code>.
240     */
241     public static Number[][] createNumberArray2D(double[][] data) {
242         ParamChecks.nullNotPermitted(data, "data");
243         int l1 = data.length;
244         Number[][] result = new Number[l1][];
245         for (int i = 0; i < l1; i++) {
246             result[i] = createNumberArray(data[i]);
247         }
248         return result;
249     }
250
251     /**
252     * Returns a {@link KeyedValues} instance that contains the cumulative
```

```

253     * percentage values for the data in another {@link KeyedValues} instance.
254     * <p>
255     * The percentages are values between 0.0 and 1.0 (where 1.0 = 100%).
256     *
257     * @param data the data (<code>>null</code> not permitted).
258     *
259     * @return The cumulative percentages.
260     */
261     public static KeyedValues getCumulativePercentages(KeyedValues data) {
262 1         ParamChecks.nullNotPermitted(data, "data");
263 1         DefaultKeyedValues result = new DefaultKeyedValues();
264 1         double total = 0.0;
265 8         for (int i = 0; i < data.getItemCount(); i++) {
266 1             Number v = data.getValue(i);
267 3             if (v != null) {
268 2                 total = total + v.doubleValue();
269             }
270         }
271 1         double runningTotal = 0.0;
272 8         for (int i = 0; i < data.getItemCount(); i++) {
273 1             Number v = data.getValue(i);
274 3             if (v != null) {
275 2                 runningTotal = runningTotal + v.doubleValue();
276             }
277 4             result.addValue(data.getKey(i), new Double(runningTotal / total));
278         }
279 1         return result;
280     }
281
282 }

```

Mutations

1. negated conditional → KILLED
- [75](#) 2. removed conditional - replaced equality check with false → KILLED
3. removed conditional - replaced equality check with true → KILLED
1. Substituted 1 with 0 → KILLED
2. Substituted 0 with 1 → KILLED
3. negated conditional → KILLED
- [76](#) 4. removed conditional - replaced equality check with false → KILLED
5. removed conditional - replaced equality check with true → KILLED
6. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
7. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
1. negated conditional → KILLED
- [78](#) 2. removed conditional - replaced equality check with false → KILLED
3. removed conditional - replaced equality check with true → KILLED
- [79](#) 1. Substituted 0 with 1 → KILLED
2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
1. negated conditional → KILLED
- [81](#) 2. removed conditional - replaced equality check with false → SURVIVED
3. removed conditional - replaced equality check with true → KILLED
- [82](#) 1. Substituted 0 with 1 → KILLED
2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
1. changed conditional boundary → KILLED
2. Changed increment from 1 to -1 → KILLED
3. Substituted 0 with 1 → SURVIVED
- [84](#) 4. negated conditional → KILLED
5. removed conditional - replaced comparison check with false → KILLED
6. removed conditional - replaced comparison check with true → KILLED
7. Removed increment 1 → TIMED_OUT
1. negated conditional → KILLED
- [85](#) 2. removed call to java/util/Arrays::equals → KILLED
3. removed conditional - replaced equality check with false → KILLED
4. removed conditional - replaced equality check with true → KILLED
- [86](#) 1. Substituted 0 with 1 → KILLED
2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
- [89](#) 1. Substituted 1 with 0 → KILLED
2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
- [102](#) 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED

1. changed conditional boundary → KILLED
 2. Changed increment from 1 to -1 → KILLED
 3. Substituted 0 with 1 → KILLED
 104 4. negated conditional → KILLED
 5. removed conditional - replaced comparison check with false → KILLED
 6. removed conditional - replaced comparison check with true → KILLED
 7. Removed increment 1 → TIMED_OUT
 1. negated conditional → KILLED
 105 2. removed conditional - replaced equality check with false → KILLED
 3. removed conditional - replaced equality check with true → KILLED
 1. Substituted 0 with 1 → KILLED
 107 2. Substituted 0 with 1 → KILLED
 3. removed call to java/lang/System::arraycopy → KILLED
 111 1. mutated return of Object value for org/jfree/data/DataUtilities::clone to (if (x != null) null else throw new RuntimeException) → KILLED
 124 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
 125 1. Substituted 0.0 with 1.0 → KILLED
 126 1. removed call to org/jfree/data/Values2D::getRowCount → KILLED
 1. changed conditional boundary → KILLED
 2. Changed increment from 1 to -1 → KILLED
 3. Substituted 0 with 1 → KILLED
 127 4. negated conditional → KILLED
 5. removed conditional - replaced comparison check with false → KILLED
 6. removed conditional - replaced comparison check with true → KILLED
 7. Removed increment 1 → KILLED
 128 1. removed call to org/jfree/data/Values2D::getValue → KILLED
 1. negated conditional → KILLED
 129 2. removed conditional - replaced equality check with false → KILLED
 3. removed conditional - replaced equality check with true → SURVIVED
 1. Replaced double addition with subtraction → KILLED
 130 2. removed call to java/lang/Number::doubleValue → KILLED
 133 1. replaced return of double value with -(x + 1) for org/jfree/data/DataUtilities::calculateColumnTotal → KILLED
 150 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
 151 1. Substituted 0.0 with 1.0 → KILLED
 152 1. removed call to org/jfree/data/Values2D::getRowCount → KILLED
 1. changed conditional boundary → KILLED
 2. Changed increment from 1 to -1 → KILLED
 3. Substituted 0 with 1 → KILLED
 153 4. negated conditional → KILLED
 5. removed conditional - replaced comparison check with false → KILLED
 6. removed conditional - replaced comparison check with true → KILLED
 7. Removed increment 1 → KILLED
 1. changed conditional boundary → KILLED
 2. negated conditional → KILLED
 155 3. removed conditional - replaced comparison check with false → KILLED
 4. removed conditional - replaced comparison check with true → KILLED
 156 1. removed call to org/jfree/data/Values2D::getValue → KILLED
 1. negated conditional → KILLED
 157 2. removed conditional - replaced equality check with false → KILLED
 3. removed conditional - replaced equality check with true → KILLED
 1. Replaced double addition with subtraction → KILLED
 158 2. removed call to java/lang/Number::doubleValue → KILLED
 162 1. replaced return of double value with -(x + 1) for org/jfree/data/DataUtilities::calculateColumnTotal → KILLED
 175 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
 176 1. Substituted 0.0 with 1.0 → KILLED
 177 1. removed call to org/jfree/data/Values2D::getColumnCount → KILLED
 1. changed conditional boundary → KILLED
 2. Changed increment from 1 to -1 → KILLED
 3. Substituted 0 with 1 → KILLED
 178 4. negated conditional → KILLED
 5. removed conditional - replaced comparison check with false → KILLED
 6. removed conditional - replaced comparison check with true → KILLED
 7. Removed increment 1 → KILLED
 179 1. removed call to org/jfree/data/Values2D::getValue → KILLED
 1. negated conditional → KILLED
 180 2. removed conditional - replaced equality check with false → KILLED
 3. removed conditional - replaced equality check with true → KILLED
 1. Replaced double addition with subtraction → KILLED
 181 2. removed call to java/lang/Number::doubleValue → KILLED
 184 1. replaced return of double value with -(x + 1) for org/jfree/data/DataUtilities::calculateRowTotal → KILLED
 201 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED

- [202](#) 1. Substituted 0.0 with 1.0 → KILLED
- [203](#) 1. removed call to org/jfree/data/Values2D::getColumnCount → KILLED
 - 1. changed conditional boundary → KILLED
 - 2. Changed increment from 1 to -1 → KILLED
 - 3. Substituted 0 with 1 → KILLED
- [204](#) 4. negated conditional → KILLED
 - 5. removed conditional - replaced comparison check with false → KILLED
 - 6. removed conditional - replaced comparison check with true → KILLED
 - 7. Removed increment 1 → KILLED
- [206](#) 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
- [207](#) 1. removed call to org/jfree/data/Values2D::getValue → KILLED
 - 1. negated conditional → KILLED
- [208](#) 2. removed conditional - replaced equality check with false → KILLED
 - 3. removed conditional - replaced equality check with true → KILLED
- [209](#) 1. Replaced double addition with subtraction → KILLED
 - 2. removed call to java/lang/Number::doubleValue → KILLED
- [213](#) 1. replaced return of double value with -(x + 1) for org/jfree/data/DataUtilities::calculateRowTotal → KILLED
- [225](#) 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
 - 1. changed conditional boundary → KILLED
 - 2. Changed increment from 1 to -1 → KILLED
 - 3. Substituted 0 with 1 → KILLED
- [227](#) 4. negated conditional → KILLED
 - 5. removed conditional - replaced comparison check with false → KILLED
 - 6. removed conditional - replaced comparison check with true → KILLED
 - 7. Removed increment 1 → TIMED_OUT
- [228](#) 1. removed call to java/lang/Double::<init> → KILLED
- [230](#) 1. mutated return of Object value for org/jfree/data/DataUtilities::createNumberArray to (if (x != null) null else throw new RuntimeException) → KILLED
- [242](#) 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
 - 1. changed conditional boundary → KILLED
 - 2. Changed increment from 1 to -1 → KILLED
 - 3. Substituted 0 with 1 → SURVIVED
- [245](#) 4. negated conditional → SURVIVED
 - 5. removed conditional - replaced comparison check with false → KILLED
 - 6. removed conditional - replaced comparison check with true → SURVIVED
 - 7. Removed increment 1 → TIMED_OUT
- [246](#) 1. removed call to org/jfree/data/DataUtilities::createNumberArray → SURVIVED
- [248](#) 1. mutated return of Object value for org/jfree/data/DataUtilities::createNumberArray2D to (if (x != null) null else throw new RuntimeException) → KILLED
- [262](#) 1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → KILLED
- [263](#) 1. removed call to org/jfree/data/DefaultKeyedValues::<init> → SURVIVED
- [264](#) 1. Substituted 0.0 with 1.0 → SURVIVED
 - 1. changed conditional boundary → KILLED
 - 2. Changed increment from 1 to -1 → SURVIVED
 - 3. Substituted 0 with 1 → SURVIVED
- [265](#) 4. negated conditional → KILLED
 - 5. removed call to org/jfree/data/KeyedValues::getItemCount → SURVIVED
 - 6. removed conditional - replaced comparison check with false → KILLED
 - 7. removed conditional - replaced comparison check with true → SURVIVED
 - 8. Removed increment 1 → SURVIVED
- [266](#) 1. removed call to org/jfree/data/KeyedValues::getValue → NO_COVERAGE
 - 1. negated conditional → NO_COVERAGE
- [267](#) 2. removed conditional - replaced equality check with false → NO_COVERAGE
 - 3. removed conditional - replaced equality check with true → NO_COVERAGE
- [268](#) 1. Replaced double addition with subtraction → NO_COVERAGE
 - 2. removed call to java/lang/Number::doubleValue → NO_COVERAGE
- [271](#) 1. Substituted 0.0 with 1.0 → SURVIVED
 - 1. changed conditional boundary → KILLED
 - 2. Changed increment from 1 to -1 → SURVIVED
 - 3. Substituted 0 with 1 → SURVIVED
- [272](#) 4. negated conditional → KILLED
 - 5. removed call to org/jfree/data/KeyedValues::getItemCount → SURVIVED
 - 6. removed conditional - replaced comparison check with false → KILLED
 - 7. removed conditional - replaced comparison check with true → SURVIVED
 - 8. Removed increment 1 → SURVIVED
- [273](#) 1. removed call to org/jfree/data/KeyedValues::getValue → NO_COVERAGE
- [274](#) 1. negated conditional → NO_COVERAGE
 - 2. removed conditional - replaced equality check with false → NO_COVERAGE

```

3. removed conditional - replaced equality check with true → NO_COVERAGE
275 1. Replaced double addition with subtraction → NO_COVERAGE
2. removed call to java/lang/Number::doubleValue → NO_COVERAGE
1. removed call to java/lang/Double::<init> → NO_COVERAGE
277 2. Replaced double division with multiplication → NO_COVERAGE
3. removed call to org/jfree/data/KeyedValues::getKey → NO_COVERAGE
4. removed call to org/jfree/data/DefaultKeyedValues::addValue → NO_COVERAGE
279 1. mutated return of Object value for org/jfree/data/DataUtilities::getCumulativePercentages to ( if (x != null)
null else throw new RuntimeException ) → SURVIVED

```

Active mutators

- RETURN_VALS_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_61
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_60
- CONDITIONALS_BOUNDARY_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_56
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_55
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_58
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_57
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_52
- VOID_METHOD_CALL_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_51
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_54
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_53
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_59
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_50
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_45
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_44
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_47
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_46
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_41
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_40
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_43
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_42
- NEGATE_CONDITIONALS_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_49
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_48
- INLINE_CONSTANT_MUTATOR
- CONSTRUCTOR_CALL_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_34
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_33
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_36
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_35
- EXPERIMENTAL_MEMBER_VARIABLE_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_30
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_32
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_31
- REMOVE_CONDITIONALS_ORDER_ELSE_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_38
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_37
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_39
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_3
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_2
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_1
- INVERT_NEGS_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_0
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_23
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_22
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_25
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_9
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_24
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_8
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_7
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_6
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_21
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_5
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_20
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_4
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_27
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_26
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_29
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_28
- REMOVE_INCREMENTS_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_12
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_11
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_99
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_14
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_13
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_96
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_95
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_10
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_98
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_97
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_19
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_16
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_15
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_18

- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_17
- EXPERIMENTAL_SWITCH_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_92
- ARGUMENT_PROPAGATION_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_91
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_94
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_93
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_90
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_89
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_88
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_85
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_84
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_87
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_86
- MATH_MUTATOR
- NON_VOID_METHOD_CALL_MUTATOR
- REMOVE_CONDITIONALS_EQUAL_IF_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_81
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_80
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_83
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_82
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_78
- REMOVE_CONDITIONALS_EQUAL_ELSE_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_77
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_79
- INCREMENTS_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_74
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_73
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_76
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_75
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_70
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_72
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_71
- REMOVE_CONDITIONALS_ORDER_IF_MUTATOR
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_67
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_66
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_69
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_68
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_63
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_62
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_65
- EXPERIMENTAL_REMOVE_SWITCH_MUTATOR_64

Tests examined

- org.jfree.data.test.DataUtilitiesCalculateRowTotalTest.calculateRowTotal_TableValuesNullTest(org.jfree.data.test.DataUtilitiesCalculateRowTotalTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCalculateColumnTotalTest.calculateColumnTotalNullMutantTest(org.jfree.data.test.DataUtilitiesCalculateColumnTotalTest) (0 ms)
- org.jfree.data.test.createNumberArrayTestMethodDataU.createNumberArray_NullMutantTest(org.jfree.data.test.createNumberArrayTestMethodDataU) (0 ms)
- org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols.calculateRowTotal_NothingNull(org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols) (1 ms)
- org.jfree.data.test.DataUtilitiesGetCumulativePercentagesTest.getCumulativePercentagesEmptyArray(org.jfree.data.test.DataUtilitiesGetCumulativePercentagesTest) (10 ms)
- org.jfree.data.test.DataUtilitiesCalculateRowTotalTest.calculateRowTotal_IsNull_LessThanZeroTest(org.jfree.data.test.DataUtilitiesCalculateRowTotalTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCloneTest.sourceElementIsNULL(org.jfree.data.test.DataUtilitiesCloneTest) (3 ms)
- org.jfree.data.test.DataUtilitiesCloneTest.sourceIsNullMutationTest(org.jfree.data.test.DataUtilitiesCloneTest) (1 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAandBPartiallyEqual(org.jfree.data.test.DataUtilitiesEqualTest) (1 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAIsNotNullBIsNullTest(org.jfree.data.test.DataUtilitiesEqualTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCloneTest.sourceIsValid(org.jfree.data.test.DataUtilitiesCloneTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest.calculateColumnTotalDataIsNULL(org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest) (51 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAandBCompletelyNotEqual(org.jfree.data.test.DataUtilitiesEqualTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCloneTest.sourceIsNull(org.jfree.data.test.DataUtilitiesCloneTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest.calculateColumnTotalDataIsNULLMutantTest(org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest) (1 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAIsNullBIsNullTest(org.jfree.data.test.DataUtilitiesEqualTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCreateNumArray2dTest.createNumberArray2D_NullMutantTest(org.jfree.data.test.DataUtilitiesCreateNumArray2dTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest.calculateColumnTotalWithAppropriateValues2D(org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCreateNumArray2dTest.createNumberArray2D_MutantTest2(org.jfree.data.test.DataUtilitiesCreateNumArray2dTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols.calculateRowTotal_NullWithinRange(org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols) (1 ms)
- org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols.calculateRowTotal_NullExact(org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols) (1 ms)
- org.jfree.data.test.DataUtilitiesCreateNumArray2dTest.createNumberArray2D_1RowTest(org.jfree.data.test.DataUtilitiesCreateNumArray2dTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCreateNumArray2dTest.createNumberArray2D_FullyPopulatedTest(org.jfree.data.test.DataUtilitiesCreateNumArray2dTest) (0 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAandBNotEqualLengths(org.jfree.data.test.DataUtilitiesEqualTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCalculateRowTotalTest.calculateRowTotal_NotNullTest(org.jfree.data.test.DataUtilitiesCalculateRowTotalTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCreateNumArray2dTest.createNumberArray2D_NormalTest(org.jfree.data.test.DataUtilitiesCreateNumArray2dTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest.calculateColumnTotalRowLessThanRowCount(org.jfree.data.test.DataUtilitiesCalculateColumnTotal3ArgsTest) (1 ms)
- org.jfree.data.test.createNumberArrayTestMethodDataU.createNumberArrayInputNull(org.jfree.data.test.createNumberArrayTestMethodDataU) (0 ms)
- org.jfree.data.test.DataUtilitiesCreateNumArray2dTest.createNumberArray2D_EmptyTest(org.jfree.data.test.DataUtilitiesCreateNumArray2dTest) (1 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAandBCompletelyEqual(org.jfree.data.test.DataUtilitiesEqualTest) (0 ms)
- org.jfree.data.test.createNumberArrayTestMethodDataU.createNumberArrayInputNotNull(org.jfree.data.test.createNumberArrayTestMethodDataU) (1 ms)
- org.jfree.data.test.DataUtilitiesEqualTest.equalAIsNullBIsNotNullTest(org.jfree.data.test.DataUtilitiesEqualTest) (0 ms)
- org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols.calculateRowTotal_NullMutantTest(org.jfree.data.test.DataUtilitiesCalculateRowTotalTestCols) (1 ms)
- org.jfree.data.test.DataUtilitiesGetCumulativePercentagesTest.getCumulativePercentages_NullMutantTest(org.jfree.data.test.DataUtilitiesGetCumulativePercentagesTest) (1 ms)
- org.jfree.data.test.DataUtilitiesCalculateColumnTotalTest.calculateColumnTotalForTwoValues(org.jfree.data.test.DataUtilitiesCalculateColumnTotalTest) (2 ms)