Course Title: Software Testing, Reliability, and Quality

Course Code: SENG 438

Assignment #: 1

Student Names:

Aashik Ilangovan (30085993)

Emmanuel Omari-Osei (30092729)

Gibran Akmal (30094918)

Priyanka Gautam (30091244)

Group Number: 31

Submission Date: 28/01/2022

Decision Tables

Data Utilities Test Method: calculateRowTotal()

Figure 1.0 - Data Utilities Class - Calculate Row Total Decision Table

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Values2D data is NOT NULL	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
Values2D data is NULL	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
int row > 0	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
int row <= 0	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N

Figure 1.1 - Data Utilities Class - Calculate Row Total Collapsed Decision Table

Conditions	1	2	3	4
Values2D data is NOT NULL	Y	Y	N	N
Values2D data is NULL	N	N	Y	Y
int row > 0	Y	N	Y	N
int row <= 0	N	Y	N	Y

Data Utilities Test Method: createNumberArray()

Figure 1.2 - Data Utilities Class - Create Number Array Decision Table

Conditions	1	2	3	4
Input Array Empty	Y	Y	N	N
Input Array Not Empty	Y	N	Y	N

Figure 1.3 - Data Utilities Class - Create Number Array Collapsed Decision Table

Conditions	1	2
**Input Array Empty	Y	N
Input Array Not Empty	N	Y

Data Utilities Test Method - getCumulativePercentages(KeyedValues data)

Figure 1.4 - Data Utilities Class - Get Cumulative Percentages Decision Table

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
data == null	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
data KeyedValues list is empty	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
all data key values are zeroes	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
only one key-value item in data	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N

Figure 1.5 - Data Utilities Class - Get Cumulative Percentages Collapsed Decision Table

Conditions	1	2	3	4	5	6
data == null	Y	N	N	N	N	N
data KeyedValues list is empty	Y	Y	N	N	N	N
all data key values are zeroes	N	N	Y	Y	N	N
only one key-value item in data	N	N	Y	N	Y	N

Data Utilities Test Method: createNumberArray2D(double[][] data)

Figure 1.6 - Data Utilities - Create Number Array 2D Decision Table

Conditions	1	2	3	4	5	6	7	8*	9	10	11	12*	13	14*	15*	16*
data == null	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
data 2d array is empty	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
Data array is 1d instead of 2d	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
Data 2d array is fully populated	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N

Figure 1.7 - Data Utilities - Create Number Array 2D Collapsed Decision Table

Conditions	1	2	3	4	5
data == null	Y	N	N	N	N
data 2d array is empty	N	Y	N	N	N
Data array is 1d instead of 2d	N	N	Y	N	N
Data 2d array is fully populated	N	N	N	Y	N

Data Utilities Test Method: calculateColumnTotal(Values2D values, int column)

Figure 1.8 - Data Utilities Class - Calculate Column Total Decision Table

Conditions	1	2	3	4	5	6	7	8
values == null	Y	Y	Y	Y	N	N	N	N
column < 0	Y	Y	N	N	Y	Y	N	N
column exists in values	Y	N	Y	N	Y	N	Y	N

Figure 1.9 - Data Utilities Class - Calculate Column Total Collapsed Decision Table

Conditions	1	2	3	4
values == null	Y	N	N	N
column < 0	N	Y	N	N
column exists in values	N	Y	Y	N

Range Test Method: getLowerBound()

Figure 2.0 - Range Class - Get Lower Bound Decision Table

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Positive Lower Bound	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
Negative Lower Bound	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
Positive Upper Bound	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
Negative Upper Bound	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N

Figure 2.1 - Range Class - Get Lower Bound Collapsed Decision Table

Conditions	1	2	3
Positive Lower Bound	Y	N	N
Negative Lower Bound	N	Y	Y
Positive Upper Bound	Y	Y	N
Negative Upper Bound	N	N	Y

Range Test Method: constrain(double value)

Fig. 2.2 - Range Class - Constrain Decision Table

Conditions	1	2	3	4
Value within Range	Y	Y	N	N
Value out of range	Y	N	Y	N

Fig. 2.3 - Range Class - Constrain Collapsed Decision Table

Conditions	1	2
Value Within Range	Y	N
Value Out of Range	N	Y

Range Test: contains(double value)

Fig. 2.4 - Range Class - Contains Decision Table

Conditions	1	2	3	4
Value within Range	Y	Y	N	N
Value out of range	Y	N	Y	N

Fig. 2.5 - Range Class - Contains Collapsed Decision Table

Conditions	1	2
Value Within Range	Y	N
Value Out of Range	N	Y

Test Method: intersects(double lower, double upper)

Figure 2.6 - Range Class - Intersects Decision Table

Conditions	1	2	3	4	5	6	7	8
Range within bound	Y	Y	Y	Y	N	N	N	N
upper bound < lower bound	Y	Y	N	N	Y	Y	N	N
lower bound == upper bound	Y	N	Y	N	Y	N	Y	N

Figure 2.7 - Range Class - Intersects Collapsed Decision Table

Conditions	1	2	3	4	5	6
Range within bound	Y	Y	Y	N	N	N
lower bound < upper bound	Y	N	N	Y	N	N
lower bound == upper bound	N	Y	N	N	Y	N

Range Test - getUpperBoundRange()

Figure 2.8 - Range Class - Get Upper Bound Range Decision Table

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Positive Lower Range	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
Negative Lower Range	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
Positive Upper Range	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
Negative Upper Range	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N

Figure 2.9 - Range Class - Get Upper Bound Range Collapsed Decision Table

Conditions	1	2	3
Positive Lower Range	Y	N	N
Negative Lower Range	N	Y	Y
Positive Upper Range	Y	Y	N
Negative Upper Range	N	N	Y