Getlowerbound:

| **Test Case Description** | **Equivalence Class** |
| --- | --- |
| **testGetLowerBound** | Range with lower bound = 0 and upper bound > 0 |
| **testGetLowerBoundForNegativeLowerBound** | Range with lower bound < 0 and upper bound > 0 |
| **testGetLowerBoundForZeroLowerBound** | Range with lower bound = 0 and upper bound > 0 |
| **testGetLowerBoundForUpperBoundEqualToLowerBound** | Range with lower bound = upper bound |
| **testGetLowerBoundForPositiveLowerBound** | Range with lower bound > 0 and upper bound > 0 |

Getlength:

|  |  |
| --- | --- |
| **testGetLength** | Range with lower bound = 0 and upper bound > 0 |
| **testGetLengthForNegativeRange** | Range with lower bound < 0 and upper bound < 0 |
| **testGetLengthForZeroRange** | Range with lower bound = 0 and upper bound = 0 |
| **testGetLengthForUpperBoundEqualToLowerBound** | Range with lower bound = upper bound |
| **testGetLengthForPositiveRange** | Range with lower bound > 0 and upper bound > 0 |

Range Constructor:

|  |  |
| --- | --- |
| **testValidRange** | Valid range with a lower bound less than the upper bound |
| **testInvalidRangeWithLowerBoundGreaterThanUpperBound** | Invalid range with a lower bound greater than the upper bound |
| **testInvalidRangeWithLowerBoundAsNaN** | Invalid range with a NaN lower bound |
| **testInvalidRangeWithUpperBoundAsNaN** | Invalid range with a NaN upper bound |
| **testInvalidRangeWithLowerBoundAsPositiveInfinity** | Invalid range with a positive infinity lower bound |
| **testInvalidRangeWithUpperBoundAsPositiveInfinity** | Invalid range with a positive infinity upper bound |
| **testInvalidRangeWithLowerBoundAsNegativeInfinity** | Invalid range with a negative infinity lower bound |
| **testInvalidRangeWithUpperBoundAsNegativeInfinity** | Invalid range with a negative infinity upper bound |

Get CentralValue

|  |  |
| --- | --- |
| testGetCentralValueRange1 | Non-empty range with odd number of elements |
| testGetCentralValueRange2 | Non-empty range with even number of elements |
| testGetCentralValueRange3 | Single element range |

Contains:

|  |  |
| --- | --- |
| testContainsInRange1 | Value within range |
| testContainsOutOfRange1 | Value outside range |
| testContainsInRange2 | Value within range |
| testContainsOutOfRange2 | Value outside range |
| testContainsSingleElementRange | Value within single element range |
| testContainsValueBelowRange | Value below range |
| testContainsValueAboveRange | Value above range |
| **Equivalence class (down below)** | **Test name (down below)** |
| Value within the range | testContainsWithinRange |
| Value outside the range | testContainsOutsideRange |
| Value on the lower bound | testContainsOnLowerBound |
| Value on the upper bound | testContainsOnUpperBound |
| Value on NaN range | testContainsOnNanRange |
| Value on lower bound of NaN range | testContainsOnLowerBoundOfNanRange |
| Value on upper bound of NaN range | testContainsOnUpperBoundOfNanRange |

Constrain:

|  |  |
| --- | --- |
| testConstrainInRange1 | value within range [1, 10] |
| testConstrainOutOfRange1Lower | value below range [1, 10] |
| testConstrainOutOfRange1Upper | value above range [1, 10] |
| testConstrainInRange2 | value within range [-5, 5] |
| testConstrainOutOfRange2Lower | value below range [-5, 5] |
| testConstrainOutOfRange2Upper | value above range [-5, 5] |
| testConstrainInRange3 | single value range [0, 0] |
| testConstrainOutOfRange3 | value outside |

expandtoInclude:

|  |  |
| --- | --- |
| testExpandToInclude1 | Range with non-zero lower and upper bounds |
| testExpandToInclude2 | Value is outside the original range |
| testExpandToInclude3 | Single value range |
| testExpandToInclude4 | Value is lower than the original lower bound |
| testExpandToInclude5 | Null input range |
| testExpandToInclude6 | Exception for null input range (not applicable) |

Hashcode:

|  |  |
| --- | --- |
| Non-null range | **testHashCode1** |
| Null range | **testHashCode2** |

Equals:

|  |  |
| --- | --- |
| Same range object | testEquals1 |
| Same range values | testEquals2, 4-7 |
| Different upper bound | testEquals3, 8 |
| Comparing to null object | testEquals9, 10 |

Shift:

|  |  |  |
| --- | --- | --- |
| C1 | Base range is null | **testShift\_NullBase** |
| EC2 | Delta is zero | **testShift\_ZeroDelta** |
| EC3 | Delta is positive | **testShift\_PositiveDelta\_AllowCrossing**, **testShift\_PositiveDelta\_NoCrossing**, **testShift\_PositiveDelta\_Crossing**, **testShift\_PositiveDelta\_CrossingNegativeToPositive**, **testShift\_PositiveDelta\_CrossingPositiveToNegative**, **testShift\_PositiveDelta\_MaxValue\_NoCrossing**, **testShift\_PositiveDelta\_MinValue\_NoCrossing**, **testShift\_PositiveDelta\_LargeRange\_NoCrossing** |
| EC4 | Delta is positive and allowZeroCrossing is true | **testShift\_PositiveDelta\_AllowCrossing**, **testShift\_PositiveDelta\_MinValue\_AllowCrossing**, **testShift\_PositiveDelta\_LargeRange\_AllowCrossing** |
| EC5 | Delta is positive and allowZeroCrossing is false | **testShift\_PositiveDelta\_NoCrossing**, **testShift\_PositiveDelta\_Crossing**, **testShift\_PositiveDelta\_CrossingNegativeToPositive**, **testShift\_PositiveDelta\_CrossingPositiveToNegative**, **testShift\_PositiveDelta\_MinValue\_NoCrossing**, **testShift\_PositiveDelta\_LargeRange\_NoCrossing** |

Shift(2):

|  |  |
| --- | --- |
| Base range is not null | **testShift\_PositiveDelta**, **testShift\_ZeroDelta1**, **testShift\_NegativeDelta** |
| Delta value is positive | **testShift\_PositiveDelta** |
| Delta value is zero | **testShift\_ZeroDelta** |
| Delta value is negative | **testShift\_NegativeDelta** |

Expand:

isNaNrange:

scale:

|  |  |
| --- | --- |
| Base range is null | **testScaleWithNullBase** |
| Scaling factor is negative | **testScaleWithNegativeFactor** |
| Scaling factor is zero | **testScaleWithZeroFactor** |
| Scaling factor is positive | **testScaleWithPositiveFactor** |
| Tests that the range is scaled correctly if the scaling factor is a large positive value. | **testScaleWithLargeFactor** |
| Tests that the range is scaled correctly if the scaling factor is a small positive value. | **testScaleWithSmallFactor** |

combineIgnoringNaN:

|  |  |
| --- | --- |
| Both Ranges are Null | testCombineIgnoringNaNBothRangesNull() |
| Range 1 is Null | testCombineIgnoringNaNRange1Null() |
| Range 2 is Null | testCombineIgnoringNaNRange2Null() |
| No NaN Value | testCombineIgnoringNaNNoNaNValue() |
| One Range Has NaN Value | testCombineIgnoringNaNOneRangeHasNaNValue() |
| Both Ranges Have NaN Value | testCombineIgnoringNaNBothRangesHaveNaNValue() |

Intersects:

|  |  |
| --- | --- |
| Normal values intersection | testIntersectsNormal |
| No intersection | testIntersectsNoIntersection |
| Null range | testIntersectsNullRange |
| Both ranges are NaN | testIntersectsNanRanges |
| One range is NaN | testIntersectsOneNanRange |
| One range is NaN, other normal | testIntersectsOneNanOneNormalRange |