# Nullness not allowed

@exception; @throws

1. [something] be/equals null
2. [something] be equal/equivalent to null
3. [something1] or [something2] be/equals null
4. [something1] or [something2] be equal/equivalent to null
5. [something] be a null value
6. [something1] or [something2] be a null value
7. [something] **parameter** be null
8. The specified [something] be null
9. Any/none of [something] be null
10. Either/neither/any/all/both/none **parameter(s)** be/equals null
11. Either/neither/any/all/both/none **parameter(s)** be equal/equivalent null
12. Either/neither [something1] or/nor [something2] be/equals null
13. Both [something1] and [something2] be null
14. The [type] be null
15. [parameter phrase] be/equals null
16. [parameter phrase] be equal/equivalent null
17. [something]’s value be/equals null
18. [something]’s value be equal/equivalent null
19. Value of [something] be/equals null
20. Value of [something] be equal/equivalent null

@param

1. [something] can not be null
2. Non-null [something]

# Nullness allowed

@param

1. [parameter] can/could be null
2. [parameter] may be null
3. [parameter] may not be null
4. [parameter] can/could be equivalent/equal to null
5. [something] **parameter** can/could be null
6. [something] **parameter** may be null
7. [something] **parameter** can/could be equivalent/equal to null
8. Value of [something] can/could be null
9. Value of [something] may be null
10. Value of [something] can/could be equal/equivalent to null
11. [something] null be ignored
12. [something] … or null

# Type restriction

@exception; @throws

1. [something] be {not} [SpecClassName]
2. [something1] or/and [something2] be {not} [SpecClassName]
3. [something] be {not} equivalent/equal to [SpecClassName]
4. [something1] or/and [something2] be {not} equivalent/equal to [SpecClassName]
5. [something] be {not} a class derived from [SpecClassName]
6. [something1] or/and [something2] be {not} a class derived from [SpecClassName]
7. [something] be {not} an instance of [type]
8. [something1] or/and [something2] be {not} an instance of [type]
9. [something] be {not} an instance of [type1] or [type2]
10. [something1] or/and [something2] be {not} an instance of [type1] or [type2]

# Range limitation

@exception; @throws

1. [something] >/</= [value]
2. [something] be {not} less/greater/larger/equal/equivalent than/to [value]
3. [something] equals [value]
4. [something1] or/and [something2] be {not} less/greater/larger/equal/equivalent than/to [value]
5. Computing [expression] be {not} less/greater/larger/equal/equivalent than/to [value]
6. Computing either [expression1] or [expression2] be {not} less/greater/larger/equal/equivalent than [value]
7. Product/sum of [something1] and [something2] be {not} less/greater/larger/equal/equivalent than/to [value]
8. [something] be {not} negative/positive/false/true
9. [something1] or/and [something2] be {not} negative/positive/false/true
10. [something1] and [something2] be {not} the same
11. [something1] equals [something2]
12. [something] be {not} in/out of/outside of range [range value]
13. [something] be {not} in/out of bounds
14. [something] be {not} [value]
15. [range expression] (only the expression，like )
16. [something] be {not} between [value1] and [value2]
17. [something] be {not} [value set]
18. [something] be {not} one of [value set]
19. [something] be {not} one of following: [value set]
20. [something] be {not} one of supported data, which are [value set]

@param:

1. [something] can/must {not} be negative/positive/non-negative/non-positive
2. [something] must be greater/less/larger than [value]
3. [something] be greater/less/larger than [value]