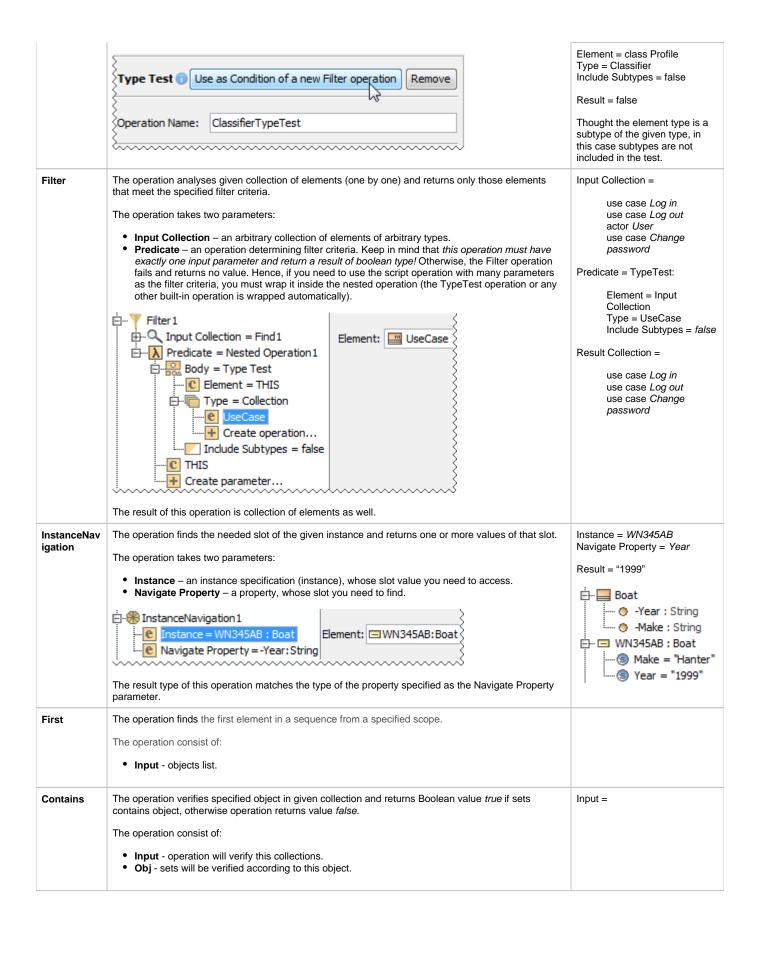
## **Built-in operations**

Operation name	Description	Example
	Date	
imestamp	The operation returns current system time in milliseconds as a string type value. It takes no parameters.	Result = "1392798008000"
FormatDate	The operation converts a date and time given in milliseconds to a human readable format.  The operation takes two parameters:  • Date – a date and time in milliseconds that should be converted to a human readable format. It must be a string type value and can be the result of a Timestamp operation.  • Format – a date and time format for the conversion. It must be a string type value. For the date and time formats, refer to the SimpleDateFormat page.  FormatDate1  Value: yyyyy.MM.dd G 'at' HH:mm:ss  The result of this operation is a string type value.	Date = "1392798008000"  Format = "yyyy.MM.dd G 'at' HH:mm:ss z"  Result = "2014.02.19 AD at 10:20:08 EET"
ParseDate	The operation converts a date and time in a human readable format to milliseconds. In other words, the operation reverses the result of the FormatDate operation (it returns the value that can be the Date parameter for a FormatDate operation).  The operation takes two parameters:  • Format – a current format of the date and time that should be converted to milliseconds. It must be a string type value.  • Formatted Date – a date and time that should be converted to milliseconds. It must be a string type value.  Date ParseDate 1  Value:  2014.02.19 AD at 10:20:08 EET  The result of this operation is a string type value.	Format = "yyyyy.MM.dd G 'at' HH:mm:ss z" Formatted Date = "2014.02.19 AD at 10:20:08 EET" Result = "1392798008000"
	Collection	
ГуреTest	The operation tests, whether the type of an element matches a given type or stereotype. If the types matches, it returns <i>true</i> , and if they not – <i>false</i> .  You can also use this operation to check, if an element is an instance of a given classifier.  The operation takes three parameters:  • Element – a model element, whose type you need to test. • Type – a type, stereotype, or classifier for testing the element. • Include Subtypes – <i>true</i> , if the inherited types, stereotypes, or classifiers of the selected Type parameter value should be included in the test; <i>false</i> , if not.  Element = Profile  TypeTest1  Element: Profile  TypeTest operation can be used in the Filter operation as the condition. For this, click the Use as Condition of a new Filter operation button on the Type Test operation specification panel.	Element = class Profile Type = Class Include Subtypes = false Result = true The element type matches the given type.  Element = class Profile Type = Classifier Include Subtypes = true Result = true Though the element type doe not match the given type, it is its subtype, and in this case subtypes are included in the test.



The operation consist of:  Input - specified object (for example array, set, list).  Size  The operation returns the number of elements in collection. The operation consist of:  Input - an object or collection (for example array, set, list, string).  Intersect  The operation returns the data common to both collections, with no repetitions and in sorted order.  The operation consist of:  Collection 1 and Collection2 - collections which will be intersected.  The operation consist of:  Collection 1 and Collection2 - collections which will be intersected.  The operation consist of:  List1 and List2 - lists which will be concatenated.  Get  The operation cretures the element at the specified position in collection. The operation returns the element or the collection of the collection. Index - index of the element to return (integer type).  Map  The operation consist of:  Input - specified collection. In experiment or even for collection to collection of other objects and returns a collection of results in the same order.  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation consist of:  Input - collections with objects to map. MapOperation - given function, this function will be applied to each input element.  MapPlat the operation consist of:  Input - collections with objects to map. MapOperation - given function, this function will be applied to each input element.  MapPlat  The operation consist of:  Input - collections with objects to map. MapOperation - given function, this function determines how collection will be reduced.  Reduce  The operation consist of:  Input - collection and will be reduced. ReduceOperation - given function, according to this function, collections will be applied. ZipOperation - given function, according to this function, collections will be applied. ZipOperation - given function, according to this function, collections will be applied. ZipOperation - given functio	IsEmpty	The operation determine whether collection is empty, if it is empty, operation returns Boolean value <i>true</i> , otherwise operation returns value <i>false</i> .	
The operation consist of:  Intersect  The operation consist of:  Intersect  The operation returns the data common to both collections, with no repetitions and in sorted order.  The operation consist of:  Collection1 and Collection2 - collections which will be intersected.  Concat  The operation joins two flists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates.  The operation consist of:  List1 and List2 - lists which will be concatenated.  Get  The operation returns the element at the specified position in collection.  The operation maps objects from collection to collection of other objects and returns a collection of returns in the collection of the collection of the collection of other objects and returns a collection of applies operation maps with function (input +1) returns = [2 3 4 5 6]  Map The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation consist of:  Input - collections of the objects from collection of other objects. Platters mapping result if it is a collection.  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  Input - collection that will be reduced.  ReduceOperation - given function, this function determines how collection will be applied.  Properation consist of:  Collection and Collection 2 to these collections, zip operation will be applied.  Zip The operation returns minimum value from given collection.		The operation consist of:	
The operation consist of:  • Input - an object or collection (for example array, set, list, string).  Intersect  The operation returns the data common to both collections, with no repetitions and in sorted order. The operation consist of:  • Collection1 and Collection2 - collections which will be intersected.  Concat  The operation pinis two lists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates. The operation consist of:  • List1 and List2 - lists which will be concatenated.  Get  The operation consist of:  • Input - specified collection. • Index - index of the element at the specified position in collection. The operation consist of:  • Input - specified collection. • Index - index of the element to return (integer type).  Map  The operation maps objects from collection to collection of other objects and returns a collection of results in the same order.  The operation consist of:  • Input - collections with objects to map. • MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  • Input - collections with objects to map. • MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation. The operation opised to the reduced. • ReduceOperation - given function, this function determines how collection will be reduced.  Input - collections that will be reduced. • ReduceOperation - given function, this function determines how collection will be reduced. • ReduceOperation - given function, according to this function, collections will be applied. • ZipOperation reduce collections using given approach to this function, collections will be zipped.		• Input - specified object (for example array, set, list).	
Intersect The operation returns the data common to both collections, with no repetitions and in sorted order. The operation consist of:  **Collection1 and Collection2 - collections which will be intersected.**  Concat  The operation joins two lists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates. The operation consist of:  **List1 and List2 - lists which will be concatenated.**  Get  The operation consist of:  **Input - specified collection.**  **Input - specified collection.**  **Input - collections with objects to map.**  **MapOperation - given function, this function will be applied to each input element.**  MapFlat  The operation consist of:  **Input - collections with objects to map.**  **Input - collection with objects to map.**  **Input - collections with objects to map.**  **Input - collection with objects to map.*	Size	The operation returns the number of elements in collection.	
Intersect  The operation returns the data common to both collections, with no repetitions and in sorted order.  The operation consist of:  * Collection1 and Collection2 - collections which will be intersected.  Concat  The operation joins two lists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates.  The operation consist of:  * List1 and List2 - lists which will be concatenated.  Get  The operation returns the element at the specified position in collection.  The operation consist of:  * Input - specified collection.  * Index - index - index of the element to return (integer type).  Map  The operation maps objects from collection to collection of other objects and returns a collection of results in the same order.  The operation consist of:  * Input - specified collections with objects to map.  * MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  * Input - collections with objects to map.  * MapOperation - given function, this function will be applied to each input element.  Reduce  The operation consist of:  * Input - collections with objects to map.  * MapOperation - given function, this function will be applied to each input element.  Reduce Operation - given function, this function determines how collection will be reduced.  * ReduceOperation - given function, this function determines how collection will be reduced.  * Poput - collection that will be reduced.  * ReduceOperation - given function, according to this function, collections will be applied.  * 2 input - collection and Collections - to these collections, zip operation will be applied.  * 2 input - collection and collections of the function, collections will be applied.  * 2 input - collection and collections of the function, collections will be applied.		The operation consist of:	
The operation consist of:  * Collection1 and Collection2 - collections which will be intersected.  Concat  The operation joins two lists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates. The operation consist of:  * List1 and List2 - lists which will be concatenated.  Get  The operation consist of:  * Input - specified collection.  * Input - specified collection.  * Index - index of the element to return (integer type).  Map  The operation consist of:  * Input - collections with objects to map.  * MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation maps objects from collection will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  * Input - collection that will be reduced.  * ReduceOperation - given function, this function determines how collection will be reduced.  Zip  The operation region consist of:  * Collection1 and Collection2 - to these collections, zip operation will be applied.  * ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collection.		• Input - an object or collection (for example array, set, list, string).	
Concat The operation joins two lists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates. The operation consist of:  List1 and List2 - lists which will be concatenated.  Get The operation returns the element at the specified position in collection. The operation consist of:  Input - specified collection. Index - index of the element to return (integer type).  Map The operation monists of:  Input - collections with objects from collection to collection of other objects and returns a collection of results in the same order. The operation consist of:  Input - operation one one of the operation of other objects. Flattens mapping result if it is a collection. The operation consist of:  Input - collections with objects to map. MapPparation - given function, this function will be applied to each input element.  Reduce The operation consist of:  Input - collection with objects to map. MapPparation - given function, this function will be applied to each input element.  Reduce The operation reduce collection using given operation. The operation reduce collections using given operation will be applied. ZipOperation reduce collections using given operation will be applied. ZipOperation reduce col	Intersect	The operation returns the data common to both collections, with no repetitions and in sorted order.	C1 = [7 0 5], C2 = [7 1 5],
Concat  The operation joins two lists. The List1 and List2 are joined end-to-end. Keeps the order, allows duplicates.  The operation consist of:  List1 and List2 - lists which will be concatenated.  Get  The operation returns the element at the specified position in collection.  The operation consist of:  Input - specified collection.  Index - index of the element to return (integer type).  Map  The operation maps objects from collection to collection of other objects and returns a collection of results in the same order.  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation reduce collections using given zip operation will be applied.  ZipOperation - given function, this function, collections will be zipped.  Min  The operation reduces collections reduced reduced.  ZipOperation and Collections? to these collections, zip operation will be zipped.		The operation consist of:	returns C = [5 7];
duplicates.  The operation consist of:  List1 and List2 - lists which will be concatenated.  Get  The operation returns the element at the specified position in collection.  The operation consist of:  Input - specified collection.  Input - specified collection to return (integer type).  Map  The operation maps objects from collection to collection of other objects and returns a collection of results in the same order.  The operation consist of:  Input - collections with objects to map.  MapPlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  Input - collections.  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  Input - collection that will be reduced.  ReduceOperation - given function, this function determines how collection will be reduced.  ReduceOperation - given function, this function determines how collection will be reduced.  Zip  The operation representation of these collections, zip operation will be applied.  ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collection.		Collection1 and Collection2 - collections which will be intersected.	
List1 and List2 - lists which will be concatenated.	Concat		
The operation returns the element at the specified position in collection.  The operation consist of:  • Input - specified collection. • Index - index of the element to return (integer type).  Map  The operation maps objects from collection to collection of other objects and returns a collection of results in the same order.  The operation consist of:  • Input - collections with objects to map. • MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  • Input - collections with objects to map. • MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  • Input - collection that will be reduced. • ReduceOperation - given function, this function determines how collection will be reduced.  The operation consist of:  • Input - collection that will be reduced. • ReduceOperation - given function, this function determines how collection will be reduced.  The operation consist of:  • Collection1 and Collection2 - to these collections, zip operation will be applied. • ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collection.		The operation consist of:	
The operation consist of:  Input - specified collection. Index - index of the element to return (integer type).  Imput - specified collection. Index - index of the element to return (integer type).  Imput = [1 2 3 4 5], applies operation map sobjects from collection to collection of other objects and returns a collection of results in the same order.  The operation consist of: Input - collections with objects to map. MapOperation - given function, this function will be applied to each input element.  Input = [2 3 4 5 6]  Input = [4.2], (3,4), (5,6) applies operation map with function (input + 1) returns = [2 3 4 5 6]  Input = (4.2), (3,4), (5,6) applies operation MapFlat returns = (1,2,3,4,5,6)  Input - collections with objects to map. MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation. The operation consist of: Input - collection that will be reduced. ReduceOperation - given function, this function determines how collection will be reduced.  Input - collection that will be reduced. ReduceOperation - given function, this function determines how collection will be reduced.  The operation consist of: Collection1 and Collection2 - to these collections, zip operation will be applied. ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.		List1 and List2 - lists which will be concatenated.	
Input - specified collection.   Index - index of the element to return (integer type).	Get	The operation returns the element at the specified position in collection.	
Index - index of the element to return (integer type).		The operation consist of:	
results in the same order.  The operation consist of:  Input - collections with objects to map.  MapPlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  Input - collections with objects to map.  MapPlat  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  The operation reduce collection using given operation.  The operation consist of:  Input - collection that will be reduced.  Reduce  The operation that will be reduced.  ReduceOperation - given function, this function determines how collection will be reduced.  ReduceOperation - given function, this function determines how collection will be reduced.  Collection1 and Collection2 - to these collections, zip operation will be applied.  ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collections.			
The operation consist of:  Input - collections with objects to map. MapOperation - given function, this function will be applied to each input element.  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of: Input - collections with objects to map. MapOperation - given function, this function will be applied to each input element.  Reduce The operation reduce collection using given operation. The operation consist of: Input - collection that will be reduced. ReduceOperation - given function, this function determines how collection will be reduced. ReduceOperation - given function, this function determines how collection will be reduced.  The operation consist of: Collection1 and Collection2 - to these collections, zip operation will be applied. ZipOperation - given function, according to this function, collections will be zipped.  The operation returns minimum value from given collections.	Мар		
• MapOperation - given function, this function will be applied to each input element.  MapFlat  The operation maps objects from collection to collection of other objects. Flattens mapping result if it is a collection.  The operation consist of:  • Input - collections with objects to map.  • MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  • Input - collection that will be reduced.  • ReduceOperation - given function, this function determines how collection will be reduced.  Zip  The operation zips two collections using given zip operation.  The operation consist of:  • Collection1 and Collection2 - to these collections, zip operation will be applied.  • ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collection.		The operation consist of:	
is a collection.  The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  Input - collection that will be reduced.  ReduceOperation - given function, this function determines how collection will be reduced.  The operation zips two collections using given zip operation.  The operation consist of:  Collection1 and Collection2 - to these collections, zip operation will be applied.  ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collection.			returns = [2 3 4 5 6]
The operation consist of:  Input - collections with objects to map.  MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  Input - collection that will be reduced. ReduceOperation - given function, this function determines how collection will be reduced.  The operation zips two collections using given zip operation.  The operation consist of:  Collection1 and Collection2 - to these collections, zip operation will be applied. ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.	MapFlat		
Input - collections with objects to map.     MapOperation - given function, this function will be applied to each input element.  Reduce  The operation reduce collection using given operation.  The operation consist of:  Input - collection that will be reduced. ReduceOperation - given function, this function determines how collection will be reduced.  The operation zips two collections using given zip operation.  The operation consist of: Collection1 and Collection2 - to these collections, zip operation will be applied. ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.		The operation consist of:	
The operation consist of:  • Input - collection that will be reduced. • ReduceOperation - given function, this function determines how collection will be reduced.  Zip  The operation zips two collections using given zip operation.  The operation consist of:  • Collection1 and Collection2 - to these collections, zip operation will be applied. • ZipOperation - given function, according to this function, collections will be zipped.  Min  The operation returns minimum value from given collection.			returns = {1,2,3,4,5,6}
Input - collection that will be reduced.     ReduceOperation - given function, this function determines how collection will be reduced.  The operation zips two collections using given zip operation.  The operation consist of:     Collection1 and Collection2 - to these collections, zip operation will be applied.     ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.	Reduce	The operation reduce collection using given operation.	
ReduceOperation - given function, this function determines how collection will be reduced.  Zip The operation zips two collections using given zip operation.  The operation consist of:      Collection1 and Collection2 - to these collections, zip operation will be applied.     ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.		The operation consist of:	
The operation consist of:  • Collection1 and Collection2 - to these collections, zip operation will be applied. • ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.		<ul> <li>Input - collection that will be reduced.</li> <li>ReduceOperation - given function, this function determines how collection will be reduced.</li> </ul>	
Collection1 and Collection2 - to these collections, zip operation will be applied.     ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.	Zip	The operation zips two collections using given zip operation.	
ZipOperation - given function, according to this function, collections will be zipped.  Min The operation returns minimum value from given collection.		The operation consist of:	
The operation consist of:	Min	The operation returns minimum value from given collection.	
		The operation consist of:	
Input - collection which will be tested and minimum value determined.		Input - collection which will be tested and minimum value determined.	

Max	The operation returns maximum value from given collection.				
	The operation consist of:				
	Input - collection which will be tested and maximum value determined.				
AllMatch	The operation checks if all collection elements match given predicate and returns boolean value.				
	The operation consist of:				
	Input - specified collection.				
	Predicate - a predicate is applied to each element of collection.				
AnyMatch	The operation checks if all collection elements match given predicate and returns boolean value.				
	The operation consist of:				
	Input - specified collection.				
	Predicate - a predicate is applied to each element of collection.				
	Logical				
And	The logical conjunction operation returns Boolean value true if all				
	parameters are true, in all other cases - operation returns false value.				
	The operation by default takes three parameters:				
	A and B - two logical values.				
	Result - a Boolean value which shows results of logical conjunction				
Or	The logical disjunction operation returns Boolean value true if either or both parameters is true,				
	otherwise operation returns <i>false</i> value.				
	The operation takes these parameters:				
	A and B - two logical values.				
Xor	The logical exclusive disjunction operation returns Boolean value <i>true</i> if both parameters differ, otherwise operation returns <i>false</i> value.				
	The operation takes these parameters:				
	A and B - two logical values.				
Not	The logical negation operation returns Boolean value <i>true</i> if parameter have value <i>false</i> , and <i>false</i> whe n parameter have value <i>true</i> .				
	The operation takes one parameter:				
	A - logical value.				
Comparison					
IfThenElse	The operation returns one or other object depending on condition.				
	The operation consist of:				
	• Condition - defines the <i>operation</i> that determines which value to assign.				
	<ul> <li>Then - expression defines the value to assign if condition is true.</li> <li>Else - expression defines the value to assign if condition is false.</li> </ul>				
LessThan	The operation returns <i>true</i> if the left parameter is less than the right parameter ( <b>A &lt; B</b> ).				
	The operation consist of:				
	A and B - parameters which will be compared.				

LessThanOr Equals	The operation returns <i>true</i> if the left side parameter is less than or equal to the right parameter (A <= B).				
	The operation consist of:				
	A and B - parameters which will be compared.				
GreaterThan	The operation returns <i>true</i> if the left parameter is greater than the right parameter ( <b>A &gt; B</b> ).				
	The operation consist of:				
	A and B - parameters which will be compared				
GreaterThan OrEquals	The operation returns <i>true</i> if the left parameter is greater than or equal to the right parameter (A >= B).				
OrEquals	The operation consist of:				
	A and B - parameters which will be compared				
Equals	The operation converts parameters if they are not the same type, then compares them. If parameters are equal, operation returns Boolean value <i>true</i> , otherwise operation returns <i>false</i> .				
	The operation consist of:				
	A and B - parameters which will be compared				
NotEquals	The operation converts parameters if they are not the same type, then compares them. If parameters are not equal, operation returns Boolean value true, otherwise operation returns false.				
	The operation consist of:				
	A and B - parameters which will be compared				
String					
StringConcat	The operation joins two parameters with string values. A and B string values are joined end-to-end.	A = Hello			
	The operation consist of:	B = world!			
	A and B - two string values.	Return = Hello world!			
StringConta ins	The operation returns Boolean value true, if specified string value is find in specified scope, otherwise.	A = 0			
IIIS	The operation consist of:	B = world!			
	<ul> <li>A - according to this string, operation will verify B</li> <li>B - string value.</li> </ul>	Return = <i>True</i>			
Other					
DiagramTyp eTest	The operation returns Boolean value true if project uses specified diagram, otherwise returns value false.	Diagram =			
	The operation consist of:	Use Case Diagram Class Diagram Free Form Diagram			
	<ul> <li>Diagram - all diagrams in the specified scope.</li> <li>Type - write diagram name exactly how it is named in modeling tool.</li> </ul>	Type = Class Diagram			
	,	Return = <i>True</i>			

## UsageInDia gramsWithS ymbols

The operation returns a collection of Diagram elements where a provided element is used in.

Searches for usages in diagrams with symbols only. Other diagrams such as Generic Tables, Dependency Matrices, Relationship Maps, etc., are not searched in. This operation performs fast and does not load any diagrams to get the information.

The operation consists of:

- element the element to search for
- predicate a predicate to filter diagrams by a custom condition. A Nested Operation must be
  the root operator for a predicate. Then, any other operation can be used inside the Nested
  Operation.

## Deprecated

## UsageInDia grams (deprecated)

Deprecated, use with caution.

This operation is very expensive. It not only checks symbol diagrams for elements but also loads and builds generic tables, dependency matrices, relationship maps, etc. It should be best avoided and used only if you understand the possible consequence of building all diagrams in the project.

Searches for usages only in all diagrams.

The operation consists of:

• element - the element to search for