

Parameter

The **Parameter** Specification window defines an operation argument.

You can [specify parameter](#) properties in the parameter [Specification window](#), which also contains descriptions.

To add additional information about the return type of an operation

1. In the **Operation** Specification window, click on the **Properties** list, and select **Expert** mode. More properties for the operation are displayed.
2. In the **Type Modifier** property specification cell, select one of the following:
 - **&** - one class has a reference to another model element.
 - ***** - one class has a pointer to another model element.
 - **[]** - one class has an array of other model elements.

To define an operation as abstract, static, or query

1. In the **Operation** Specification window, click on the **Properties** list, and select **Expert** mode. More properties for the operation are displayed.
2. In the **General** property group, select one or all of the following
 - **Is Abstract**: the operation does not have an implementation; one must be supplied by a descendant.
 - **Is Static**: the values returned by the parameter have no duplicates.
 - **Is Query**: does not change the state of the system.

To set the operation visibility

- In the **Operation** Specification window, click on the **Visibility** property specification cell, and select one of the following:
 - **Public**: the operation can be accessed by any other object from the outside.
 - **Package**: the operation can be accessed by an element from the same package.
 - **Private**: the operation can be accessed only from that class.
 - **Protected**: the operation can be accessed from the inside of that class and the classes derived from that class.

To set an operation Concurrency

1. In the **Operation** Specification window, click on the **Properties** list, and select **Expert** mode. More properties for the operation are displayed.
2. In the **Concurrency** property specification cell, select one of the following:
 - **Sequential**: The callers must coordinate, so only one call to an Instance (on any sequential Operation) is made at a time. If simultaneous calls occur, the semantics and integrity of the system cannot be guaranteed.
 - **Guarded**: Multiple calls from concurrent threads may occur simultaneously to one Instance (on any guarded Operation), but only one is allowed to commence. The others are blocked until the performance of the first Operation is complete. It is the responsibility of the system designer to ensure that deadlocks do not occur due to simultaneous blocks. The Guarded Operations must perform correctly (or block themselves) if a simultaneous sequential Operation or guarded semantics cannot be claimed.
 - **Concurrent**: Multiple calls from concurrent threads can occur simultaneously to one Instance (on any concurrent Operation). All of them can proceed concurrently with the correct semantics. The Concurrent Operations must perform correctly in case a simultaneous sequential or the guarded Operation, or concurrent semantics cannot be claimed.

Related Pages

- [Model Elements](#)
- [Specification Window](#)