cast

Syntax	`anObject = cast(anotherObject)`	
Semantics	Tells the compiler to compile the statement without checking if the variable types correspond. At runtime there are no type checks, and the statement will be executed as it is. This is useful if e.g. an interface call returns an **Any** parameter or a parameter of abstract type, but you, as a modeler, know the concrete type during development. On the other hand, this operation must be used carefully because it will never throw any error. All assignment statements will compile and will be processed at runtime - regardless of the output. This may cause follow-up errors in other sections of your model, and it may get difficult for you to identify the `cast()` as the source of this. Concerning simple type classes and types that are derived from them, `castValue()` is the better option.	
Substitutables	`anObject`	An object having a dedicated type. This type is the target type of the operation.
	`anotherObject`	An object having any type. This can be a simple type like **String**, **Integer**, **Float**, **DateTime**, or **Array**, or any complex type.
Examples	`set aConcreteParameter = cast(anAbstractParameter);`	
Error Codes	This operation must be used carefully because it will never throw any error.	

Related Pages:

- castValue() Operation

Example File (Builder project Basic Modeling/Data):

	<your example path>\Basic Modeling\Data\uml\cast.xml