## 27.8 Getting attribute values

TTCN‑3 contains a set of operations that can be used for retrieving attribute values associated with a type, template, variable, constant or module parameter.

### 27.8.1 The display of statement

***Syntactical Structure***

**display of** ( *Type* | *TemplateInstance* )

***Semantic Description***

The **display** **of** operation returns the value of the display attribute as a universal charstring. An empty string is returned if the display attribute is not defined.

### 27.8.2 The encode of statement

***Syntactical Structure***

**encode of** ( *Type* | *TemplateInstance* )

***Semantic Description***

The **encode** **of** operation returns the value of the encode attribute as a universal charstring. If more than one encode attribute is defined, the operation returns a string that contains comma-separated values of all encode attributes. An empty string is returned if the encode attribute is not defined.

### 27.8.3 The variant of statement

***Syntactical Structure***

**Variant [** "("*FreeText* ")" **] of** ( *Type* | *TemplateInstance* )

***Semantic Description***

The **variant** **of** operation returns the value of the variant attribute as a universal charstring. The variant of statement can contain an optional textual parameter specifying an encoding. If the parameter is missing, the statement return the variant value that is common for all encodings. If the parameter is present, the statement returns the variant that is valid for the specified encoding. In case more than one variant attribute is defined, the operation returns a string that contains comma-separated values of all variant attributes. An empty string is returned if the variant attribute is not defined.

### 27.8.4 The extension of statement

***Syntactical Structure***

**extension of** ( *Type* | *TemplateInstance* )

***Semantic Description***

The **extension** **of** operation returns the value of the extension attribute as a universal charstring. If more than one extension attribute is defined, the operation returns a string that contains comma-separated values of all extension attributes. An empty string is returned if the extension attribute is not defined.

### 27.8.5 The optional of statement

***Syntactical Structure***

**optional of** *TemplateInstance*

***Semantic Description***

The **optional** **of** operation returns the value of the optional attribute as a universal charstring. An empty string is returned if the optional attribute is not defined.

#### A.1.6.8.3 Basic statements

534. OpCall ::= [ConfigurationOps](#TConfigurationOps) |

 [GetLocalVerdict](#TGetLocalVerdict) |

 [TimerOps](#TTimerOps) |

 [TestcaseInstance](#TTestcaseInstance) |

 ([FunctionInstance](#TFunctionInstance) [[ExtendedFieldReference](#TExtendedFieldReference)]) |

 ([TemplateOps](#TTemplateOps) [[ExtendedFieldReference](#TExtendedFieldReference)]) |

 [ActivateOp](#TActivateOp) |

 AttributeOp

???. AttributeOp ::= AttributeSpec [OfKeyword](#TOfKeyword) ( [Type](#TType) | TemplateInstance )

???. AttributeSpec ::= [EncodeKeyword](#TEncodeKeyword) |

 [VariantKeyword](#TVariantKeyword) ["(" [FreeText](#TFreeText) ")"] |

 [DisplayKeyword](#TDisplayKeyword) |

 [ExtensionKeyword](#TExtensionKeyword) |

 [OptionalKeyword](#TOptionalKeyword)