ETSI ES 203 790 V1.2.1 (2020-05)

Methods for Testing and Specification (MTS);

The Testing and Test Control Notation version 3;

TTCN-3 Language Extensions: Object-Oriented Features

**ETSI Standard**

Reference

RES/MTS-203790-OOFv1.2.1

Keywords

language, TTCN-3

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C

Association à but non lucratif enregistrée à la

Sous-Préfecture de Grasse (06) N° 7803/88

***Important notice***

The present document can be downloaded from:  
<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:  
<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.  
The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.  
**3GPP™**and **LTE™** are trademarks of ETSI registered for the benefit of its Members and  
of the 3GPP Organizational Partners.  
**oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and  
of the oneM2M Partners.  
**GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

#### 5.1.1.11 Built-in classes

The abstract special built-in class called object is the superclass for all classes that do not explicitly extend another class.

The pseudo definition of that class is:

**type class @abstract @builtin** object {

// This function will return a tool-specific descriptive string by default

// but can be overridden by subclasses  
   **public function** toString() **return universal charstring**;

// Indicates wether some object is semantically equivalent to this one,

// according to some equivalence relation.

// The default implementation returns true if and only if both this and obj

// are the same object instance, otherwise returns false.

**public function** equals(object obj) **return boolean** {

**return** this == obj;

}

}

NOTE: The @builtin is only added for illustrative purposes and not part of the TTCN-3 language.

EXAMPLE:

**type class** Square {

**public** **function** getSideLength() **return** **integer** { … }  
…  
 **public function** equals(object obj) **return boolean** {  
 **if** (obj **of** Rectangle) {  
 // a rectangle is a suare if it has 4 sides of equal lengths  
 **var** Rectangle rectangle := obj => Rectangle;  
 **if** (rectangle.getNofSides() != 4) {  
 **return false**;  
 }

**var** **integer** tempSideLength := rectangle.getSideLength(0);  
 **return** templSideLength == getSideLength() and   
 tempSideLength == rectangle.getSideLength(1)**and**   
 tempSideLength == rectangle.getSideLength(2) **and**   
 tempSideLength == rectangle.getSideLength(3);  
 }  
 **else** **if** (obj **of** Square) {  
 **return** getSideLength() == obj=>Square.getSideLength();  
 }  
  
 **return** **this** == obj;  
 }  
}

**type class** Rectangle {  
…  
 **public function** getNofSides() **return integer** { … }  
 **public function** getSideLength(in integer index) **return integer** { … }  
 **public function** equals(object obj) **return boolean** {  
 **if** (obj **of** Square) {  
 // a square is always a rectangle  
 **return obj.equals(this);**  
 }  
 **else** if (obj **of** Rectangle) {  
 **var** Rectangle r2 := obj => Rectangle;  
 **if** (getNofSides() != r2.getNofSides()) { **return** **false** }  
 **for** (**var** **integer** i := 0; i < getNofSides(); i := i + 1) {  
 **if** (getSideLength(i) != r2.getSideLength(i)) { **return** **false** }  
 }  
 **return** **true**;  
 }  
 **return** **this** == obj;  
 }  
}