IEC TR 61131-8

Edition 3.0 2017-11

TECHNICAL REPORT

colour inside

Industrial-process measurement and control – Programmable controllers – Part 8: Guidelines for the application and implementation of programming languages

IEC TR 61131-8:2017-11(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IEC TR 61131-8

Edition 3.0 2017-11

TECHNICAL REPORT

colour inside

Industrial-process measurement and control – Programmable controllers – Part 8: Guidelines for the application and implementation of programming languages

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 25.240.50

ISBN 978-2-8322-4898-0

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

INDUSTRIAL-PROCESS MEASUREMENT AND CONTROL – PROGRAMMABLE CONTROLLERS –

Part 8: Guidelines for the application and implementation of programming languages

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC 61131-8, which is a technical report, has been prepared by subcommittee 65B: Measurement and control devices, of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition cancels and replaces the second edition published in 2003. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

This third edition is a compatible extension of the second edition. The main extensions are new data types and conversion functions, references, name spaces and the object oriented features of classes and function blocks (see listing in Annex B of IEC 61131-3:2013).

The text of this technical report is based on the following documents:

DTR	Report on voting
65B/1058/DTR	65B/1073/RVDTR

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 61131 series, published under the general title *Industrial-process measurement and control – Programmable controllers*, can be found on the IEC website.

Future standards in this series will carry the new general title as cited above. Titles of existing standards in this series will be updated at the time of the next edition.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This part of IEC 61131 is being issued as a technical report in order to provide guidelines for the implementation and application of the programming languages defined in IEC 61131-3:2013.

The content of this document answers a number of frequently asked questions about the intended application and implementation of the normative provisions of IEC 61131-3.

INDUSTRIAL-PROCESS MEASUREMENT AND CONTROL – PROGRAMMABLE CONTROLLERS –

Part 8: Guidelines for the application and implementation of programming languages

1 Scope

This part of IEC 61131, which is a technical report, applies to the programming of programmable controller systems using the programming languages defined in IEC 61131-3. The scope of IEC 61131-3 is applicable to this part.

This document provides

a) guidelines for the application of IEC 61131-3,

b) guidelines for the implementation of IEC 61131-3 languages for programmable controller systems,

c) programming and debugging tool (PADT) recommendations.

For further information IEC 61131-4 describes other aspects of the application of programmable controller systems, e.g. electromagnetic compatibility or functional safety.

NOTE Neither IEC 61131-3 nor this document explicitly addresses safety issues of programmable controller systems or their associated software. The various parts of IEC 61508 can be consulted for such considerations.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-3:2013, *Programmable controllers – Part 3: Programming languages*

IEC 61131-5, *Programmable controllers – Part 5: Communications*

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

4 Abbreviated terms

ED	Early Detection
ETrig	Edge triggered function
EW	Early Warning

FB	Function Block
FBD	Function Block Diagram
FC	Function
IL	Instruction List
LD	Ladder Diagram
MMS	Manufacturing Message Specification
NaN	Not a number
OO	Object Orientation
OOP	Object Oriented Programming
PLC	Programmable Logic Controller
POU	Program Organization Unit
PADT	Programming And Debugging Tool
RT	RunTime
SFC	Sequential Function Chart
ST	Structured Text
VMD	Virtual Manufacturing Device

5 Overview

The intended audience for this document consists of

– users of programmable controller systems as defined in IEC 61131-3, who shall program, configure, install and maintain programmable controllers as part of industrial-process measurement and control systems; and

– implementers of programming and debugging tools (PADT)as defined in IEC 61131-3, for programmable controller systems. This can include vendors of software and hardware for the preparation and maintenance of programs for these systems, as well as vendors of the programmable controller systems themselves.

IEC 61131-3 is mainly oriented toward the implementers of programming languages for programmable controllers. Users who wish a general introduction to these languages and their application should consult any of several generally available textbooks on this subject.

Clause 6 of this document provides a general introduction to IEC 61131-3, while Clause 7 provides complementary information about the application of some of the programming language elements specified in IEC 61131-3. Clause 8 provides information about the intended implementation of some of these programming language elements.

Hence, it is expected that users of programmable controllers will find Clauses 6 and 7 of this part most useful, while programming language implementers will find Clause 8 more useful, referring to the background material in Clauses 6 and 7 as necessary.

6 Introduction to IEC 61131-3

6.1 General considerations

In the time before the IEC 61131-3 was existing, the limited capabilities of expensive hardware components imposed severe constraints on the design process for industrial-process control, measurement and automation systems. Software design and implementation were tightly tailored to the selected hardware. This required specialists who were highly skilled, both in solving process automation problems and in dealing with complicated, often hardware-specific computer programming constructs.

With the rapid innovation in microelectronics and related technologies, the cost/performance ratio of system hardware has decreased dramatically. At present, a programmable controller can cost many times less than the cost of programming it.

Driven by rapidly decreasing hardware cost, a trend has become established of replacing large, centrally installed process computers or other comparatively large, isolated controllers by systems with spatially and functionally distributed parts.

As illustrated in Figure 1, the essential backbone of such systems is the communication subsystem, which provides the mechanism for information exchange between the distributed automating devices. Connected to this backbone are the devices, such as programmable controllers, which deliver the distributed processing power of the system. Each device, under the control of its own software, performs a dedicated subtask to achieve the required overall system functionality. Each device is chosen with the size and performance required to meet the demands of its particular subtask.

In a different environment, programmable controllers are used in stand-alone applications as illustrated in Figure 2. Users of these applications also stand to gain by the evolution outlined above. Due to the present low cost of hardware components, many new, relatively small, automation tasks can be solved profitably and flexibly by programmable controllers.

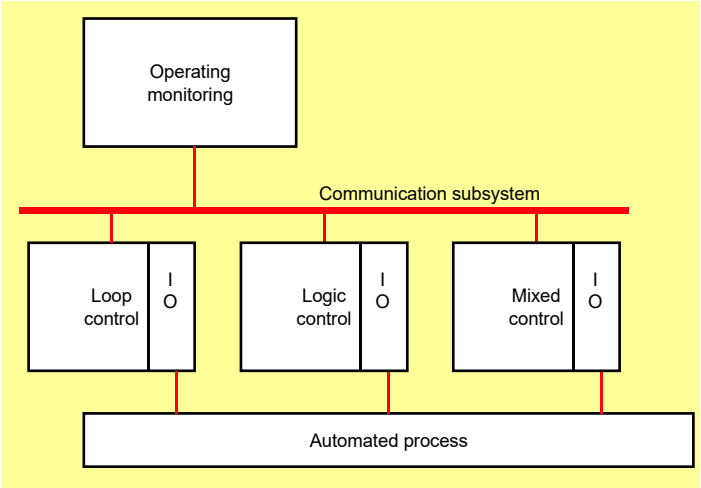
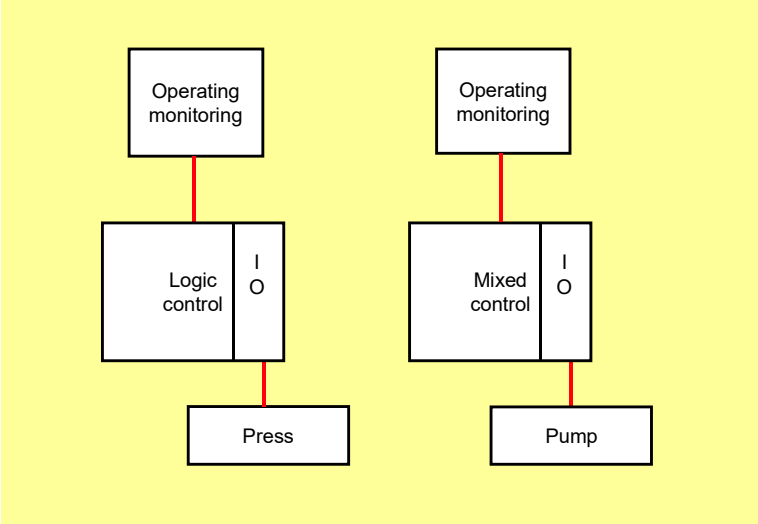


Figure 1 – A distributed application



IEC

Figure 2 – Stand-alone applications

In addition to their low hardware price, the intensive use of programmable controllers in solving automation tasks is also advanced by their straightforward operating and programming principles, which are easily understood and applied by the shop-floor personnel involved in programming, operation and maintenance.

Programmable controllers typically employ the principles of cyclic or periodic program execution illustrated in Figure 3. Cyclically running programs restart execution as fast as possible after they have terminated execution, in IEC 6113-3 such programs are not assigned to a task and executed with the lowest priority. Periodic execution of a program is triggered by a clock mechanism at equidistant points in time. The same controller may execute such programs quasi-simultaneous with different periodic times. Another kind (defined by IEC 61131-3 but not shown in Figure 3) is the non-periodic execution, which is based on events and will be executed upon each occurrence of such events. These principles are well known and applied in the operation of digital signal processing systems to simulate the operation of continuously operating analogue or electromechanical systems. Process values are read into the device and written out to the process as discrete samples at random or equidistant points in time, depending on the control task that has to be fulfilled.

The advantage of these operating principles is that they allow the construction of programs for programmable controllers using elements closely related to the principles of hard-wired logic or continuous control circuits previously used for the same purpose.

The operating principles of programmable controllers thus enable the provision of application-specific, graphical programming languages. Combined with appropriate man-machine interfaces, these languages enable the control engineer to concentrate on solving the problems of the application, without extensive training in software engineering. The control engineer's technological specifications can be mapped directly to the corresponding language elements.

Another particular advantage of such programming languages is that the representation they offer can be used not only for program input and documentation, but also for on-line test and diagnosis as well. Thus, programming and debugging tools (PADT) for programmable controllers are able to provide the graphically oriented representation and documentation that are already familiar to the application engineer and shop-floor personnel.

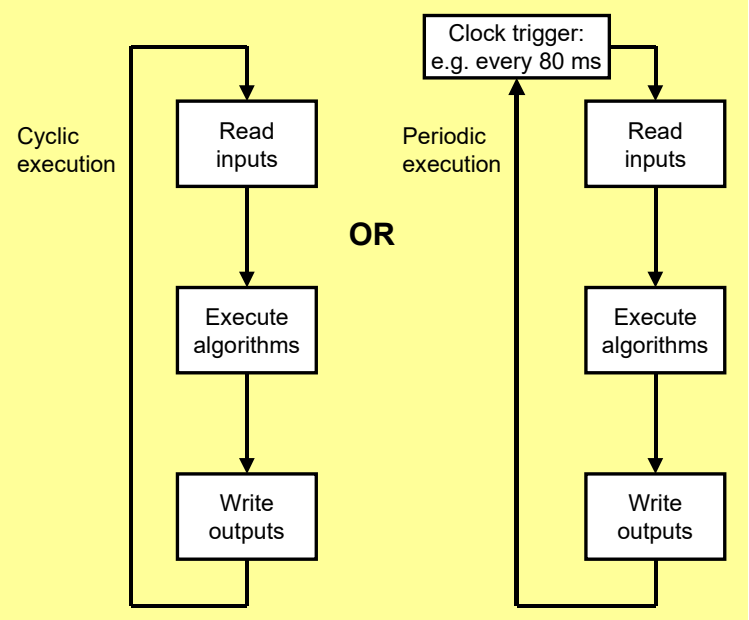


Figure 3 – Cyclic or periodic scanning of a program

6.2 Overcoming historical limitations

Automation system designers are often required to use programmable controllers from various manufacturers in different automation systems, or even in the same system. However, the hardware of programmable controllers from different manufacturers may have very little in common. This has resulted in significant differences in the elements and methods of programming the software as well and has led to the development of manufacturer-specific programming and debugging tools, which generally carried very specialized software for programming, testing and maintaining particular controller "families".

Changing from one controller family to another often required the designer to read large manuals for both the hardware and software of the new family. Often, the manual had to be reviewed several times in order to understand the exact meaning and to use the new controller family in an appropriate way. Due to the concentrated, tedious work necessary to read and understand the new, vendor-specific material, few people did it. For this reason, many people regarded the design and the programming of such controllers as some black magic to be avoided. Thus, the knowledge of how to use such systems effectively was concentrated in one or a few specialists and could not be transferred effectively to those responsible for system operation, maintenance, and upgrade.

A major goal of IEC 61131-3 was to remove such barriers to the understanding and application of programmable controllers. Thus, IEC 61131-3 introduced numerous facilities to support the advantages of programmable controllers described in 6.1, even if controllers of different vendors are concerned. It has turned out that the resulting expansion of the application domains of programmable controllers, and the increasing demand of customers fed through this expansion, stimulated a lot of vendors to make their programming systems compliant to the standard.

Vendor and user organizations like PLCopen accelerated this process by promoting the benefits and advantages of standardizing PLC programming to a large extent.

6.3 Basic features in IEC 61131-3

From the point of view of the application engineer and the control systems configurator, the most important features introduced by IEC 61131-3 can be summarized as follows.

a) Well-structured, “top-down” or “bottom-up” program development is facilitated by language constructs for the definition of typed functional objects (program organization units) such as functions, function blocks, classes and programs.

b) Strong data typing is not only supported but inherently required, thus eliminating a major source of programming errors.

c) A sufficient set of features for the execution control of program organization units is included; those features associated with steps, transitions and action blocks offer excellent means to represent complicated sequential control solutions in a concise form.

d) The functionality for designing the communication using the VAR_ACCESS keyword between application programs is provided. Independent of the mapping of programs onto a single device or different devices, identical communication features can be used between two programs. This facilitates the reuse of software in different environments.

e) Graphical or textual languages may be chosen by the designer, according to the requirements of the application. These languages, plus a set of textual and graphical common elements, support software design methodologies based on well-understood models.

   1) The graphical Ladder Diagram (LD) language models networks of simultaneously functioning electromechanical elements such as relay contacts and coils, timers, counters, etc.

   2) The graphical Function Block Diagram (FBD) language models networks of simultaneously functioning electronic elements such as adders, multipliers, shift registers, gates, etc.

   3) The Structured Text (ST) language models typical information processing tasks such as numerical algorithms using constructs found in general-purpose high level languages such as Pascal.

   4) The Instruction List (IL) language models the low-level programming of control systems in assembly language.

      In IEC 61131-3:2013, IL is marked as deprecated for the next release, because an assembler like language is not up-to-date in modern development environments.

   5) A set of graphical and textual common elements provides rules for defining values and variables, features for software configuration and object declaration. The common elements include graphical and textual elements for the construction of program organization units.

   6) Sequential Function Charts (SFCs), which model time- and event-driven sequential control devices and algorithms.

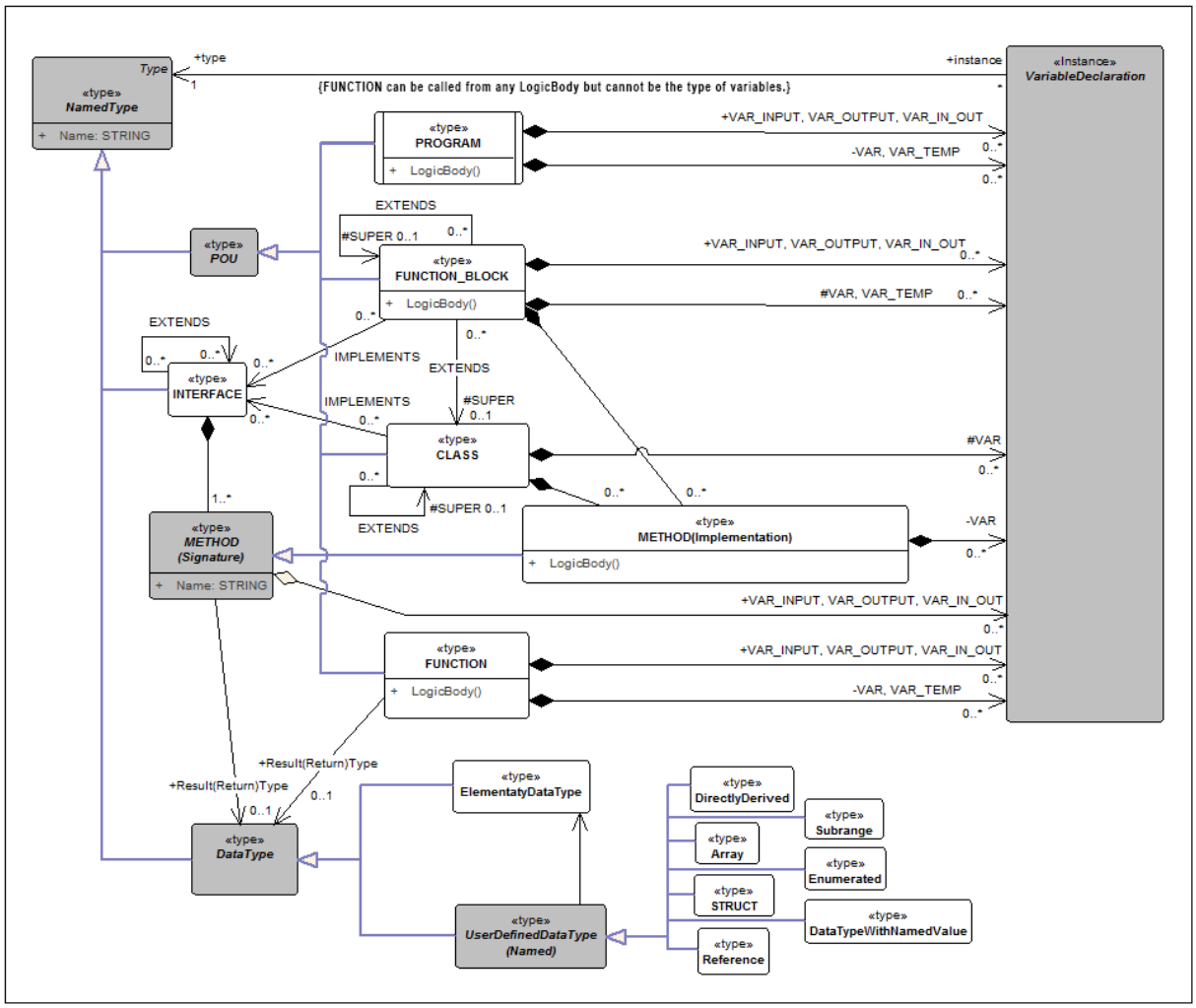
f) Flexibility in the selection of languages suited for programming application-specific functionalities will increase the reuse of software solutions to process control problems.

g) Each application specialist on a project team can use a programming style and language suited for the particular functionality in question, with the assurance that the results of the work of the individual specialists will integrate smoothly together.

In summary, the principal goal of IEC 61131-3 is to introduce all the necessary standardized language concepts and constructs to solve the technological problems of each application and to provide principles for the construction of vendor-independent software elements. This facilitates the reusability of control software designs for different controller types, even though some effort will almost always be required in order to move control programs from one controller family to another.

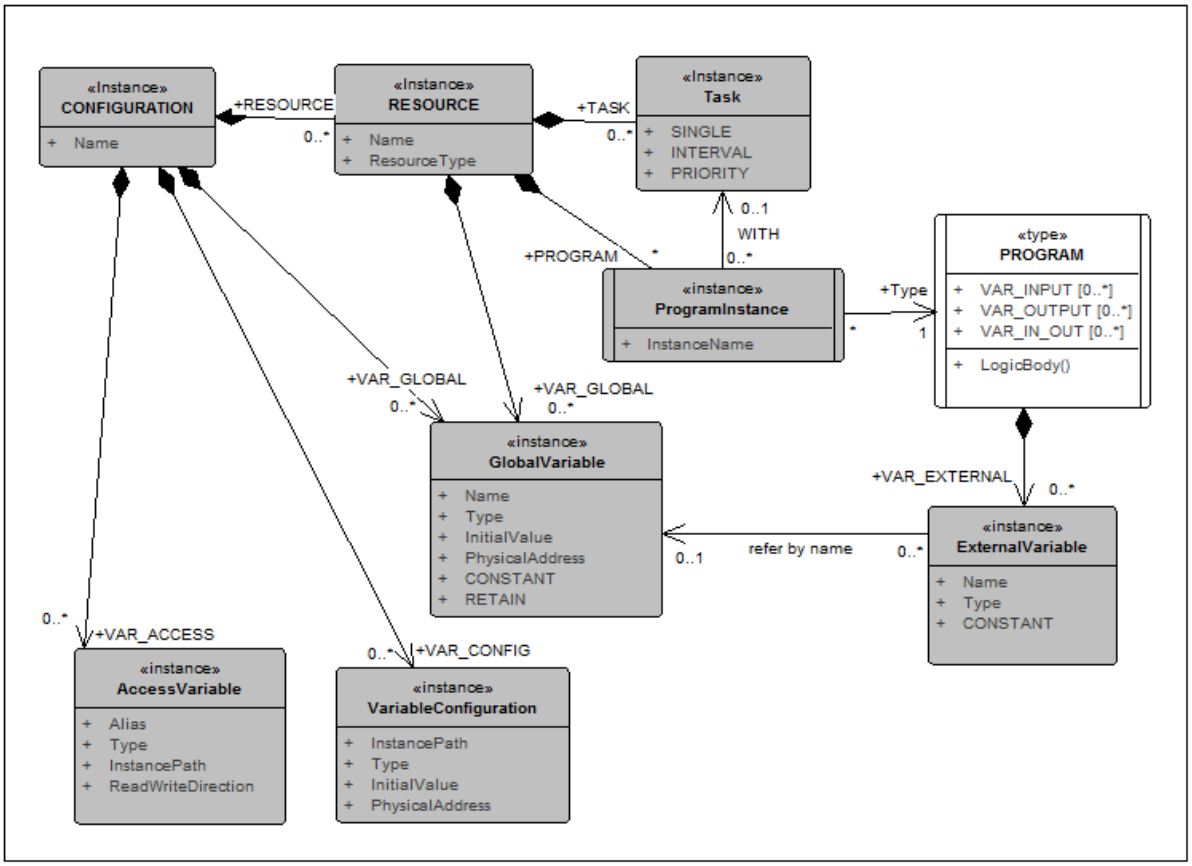
6.4 Language items overview

The basic high-level language elements for programming and their relationship are shown in the programming model shown in Figure 4 and Figure 5.



IEC

Figure 4 – Programming Modelm



IEC

Figure 5 – Software Modelm

6.5 Changes from IEC 61131-3:2003 (edition 2) to IEC 61131-3:2013 (edition 3)

One of the major changes is the extension of the basic concept of the programming language by supporting object oriented features beyond FB instantiation. Major object oriented benefits in IEC 61131-3 are:

- Better structuring of the program
- Extension by reusing existing FBs and Classes
- Maintenance and extensions

The changes made to IEC 61131-3:2003 are described in IEC 61131-3:2013, Annex B.

- Object oriented extensions
  - Introducing CLASS and METHOD.
  - Introducing INTERFACE element and IMPLEMENTS declaration by POU.
  - Overriding body and method of the base POU by EXTENDS and OVERRIDE keyword
  - Introducing access specifier PUBLIC / INTERNAL / PROTECTED / PRIVATE / FINAL for internal variables of CLASS and FUNCTION_BLOCKs
  - Extending FUNCTION_BLOCK to support these OO extensions.
- Introducing NAMESPACE
  - Declaring NAMESPACE in type definitions.
  - USING direction
  - Access specifier INTERNAL for the contents of the namespace.
- Addition of some standard data types.

  - 64 bit time types (LTIME, LTOD, LDT),
  - Character type (CHAR, WCHAR)
- Addition of some standard functions and function blocks
  - Functions to operate newly added data types above
  - Typed TRUNC, endian conversion, validation, ATAN2, etc.
- Extension of user-defined data types
  - Reference type, i.e. typed pointer without arithmetic
  - Variable-length array for the type of input / in_out variables of POU
  - Data type with named values
  - Data type with explicit layout
  - Partial access to ANY_BIT variables
- Miscellaneous
  - Addition of some commenting notations. (//, /* */, Nesting comment permitted.)
  - Addition of compare contact in ladder diagram
  - Addition of CONTINUE statement in structured text (ST) language
  - Function without result become permitted
  - Recursive call of POU become implementer specific
  - Implicit / explicit type conversion rules are clarified

6.6 Software engineering considerations

6.6.1 Application of software engineering principles

6.6.1.1 Encapsulation and hiding

A number of software engineering principles were employed in the development of IEC 61131-3 in order to promote increased software quality. A few of the more important principles, their contributions to software quality, and their embodiment in IEC 61131-3 are discussed below.

NOTE See Annex A of this document for descriptions of the software engineering quality measures referenced in 6.6.1, for example, reliability, maintainability, etc.

Encapsulation is the “packaging” of functionally related data and/or procedures in a single software entity. Encapsulation contributes to software reliability, maintainability, usability, and adaptability.

The notion of hiding of procedures and data is associated with encapsulation. "Hiding" means that the user does not get information about the internal data structure and procedure implementation of a software entity. He only gets information about its external interface and specified functionality. Hiding contributes to software maintainability, integrity, usability, portability and reusability.

IEC 61131-3 supports encapsulation and hiding by the following elements:

- Program;
- Functionblock;
- Class;
- Interface;
- Function;
- Method;
- Datatypes.