
**Information technology —
Programming languages — C#**

Technologies de l'information — Langages de programmation — C#

iTeh Standards
(<https://standards.itih.ai>)
Document Preview

[ISO/IEC 23270:2018](https://standards.itih.ai/catalog/standards/iso/546611e8-5563-4498-98c2-4c8c71c1d6be/iso-iec-23270-2018)

<https://standards.itih.ai/catalog/standards/iso/546611e8-5563-4498-98c2-4c8c71c1d6be/iso-iec-23270-2018>



iTeh Standards
(<https://standards.itih.ai>)
Document Preview

ISO/IEC 23270:2018

<https://standards.itih.ai/catalog/standards/iso/546611e8-5563-4498-98c2-4c8c71c1d6be/iso-iec-23270-2018>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2018

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Table of Contents

Foreword.....	xix
Introduction	xxi
1. Scope.....	1
2. Normative references	3
3. Terms and definitions.....	5
4. Acronyms and abbreviations	7
5. General description.....	9
6. Conformance	11
7. Lexical structure.....	13
7.1 Programs	13
7.2 Grammars.....	13
7.2.1 General.....	13
7.2.2 Grammar notation	13
7.2.3 Lexical grammar	14
7.2.4 Syntactic grammar	15
7.2.5 Grammar ambiguities	15
7.3 Lexical analysis	16
7.3.1 General.....	16
7.3.2 Line terminators.....	16
7.3.3 Comments	17
7.3.4 White space.....	18
7.4 Tokens	19
7.4.1 General.....	19
7.4.2 Unicode character escape sequences.....	19
7.4.3 Identifiers	20
7.4.4 Keywords.....	21
7.4.5 Literals	22
7.4.5.1 General.....	22
7.4.5.2 Boolean literals.....	23
7.4.5.3 Integer literals	23
7.4.5.4 Real literals.....	24
7.4.5.5 Character literals	25
7.4.5.6 String literals	26
7.4.5.7 The null literal	28
7.4.6 Operators and punctuators.....	28
7.5 Pre-processing directives	28
7.5.1 General.....	28
7.5.2 Conditional compilation symbols.....	30
7.5.3 Pre-processing expressions.....	30
7.5.4 Definition directives	31
7.5.5 Conditional compilation directives	32

7.5.6 Diagnostic directives	34
7.5.7 Region directives	35
7.5.8 Line directives	35
7.5.9 Pragma directives	36
8. Basic concepts.....	37
8.1 Application startup	37
8.2 Application termination	38
8.3 Declarations	38
8.4 Members	41
8.4.1 General	41
8.4.2 Namespace members	41
8.4.3 Struct members	41
8.4.4 Enumeration members	42
8.4.5 Class members	42
8.4.6 Interface members	42
8.4.7 Array members	42
8.4.8 Delegate members	42
8.5 Member access	42
8.5.1 General	42
8.5.2 Declared accessibility	42
8.5.3 Accessibility domains	43
8.5.4 Protected access	46
8.5.5 Accessibility constraints	47
8.6 Signatures and overloading	48
8.7 Scopes	49
8.7.1 General	49
8.7.2 Name hiding	52
8.7.2.1 General	52
8.7.2.2 Hiding through nesting	52
8.7.2.3 Hiding through inheritance	53
8.8 Namespace and type names	54
8.8.1 General	54
8.8.2 Unqualified names	56
8.8.3 Fully qualified names	56
8.9 Automatic memory management	57
8.10 Execution order	59
9. Types	61
9.1 General	61
9.2 Reference types	61
9.2.1 General	61
9.2.2 Class types	62
9.2.3 The object type	62
9.2.4 The dynamic type	63
9.2.5 The string type	63
9.2.6 Interface types	63
9.2.7 Array types	63
9.2.8 Delegate types	63
9.3 Value types	63
9.3.1 General	63

9.3.2 The System.ValueType type	64
9.3.3 Default constructors	64
9.3.4 Struct types	65
9.3.5 Simple types	65
9.3.6 Integral types	66
9.3.7 Floating-point types	67
9.3.8 The decimal type	68
9.3.9 The bool type	69
9.3.10 Enumeration types	69
9.3.11 Nullable value types	69
9.3.12 Boxing and unboxing	70
9.4 Constructed types	70
9.4.1 General	70
9.4.2 Type arguments	71
9.4.3 Open and closed types	71
9.4.4 Bound and unbound types	72
9.4.5 Satisfying constraints	72
9.5 Type parameters	73
9.6 Expression tree types	73
9.7 The dynamic type	74
10. Variables	77
10.1 General	77
10.2 Variable categories	77
10.2.1 General	77
10.2.2 Static variables	77
10.2.3 Instance variables	77
10.2.3.1 General	77
10.2.3.2 Instance variables in classes	78
10.2.3.3 Instance variables in structs	78
10.2.4 Array elements	78
10.2.5 Value parameters	78
10.2.6 Reference parameters	78
10.2.7 Output parameters	79
10.2.8 Local variables	79
10.3 Default values	80
10.4 Definite assignment	80
10.4.1 General	80
10.4.2 Initially assigned variables	81
10.4.3 Initially unassigned variables	81
10.4.4 Precise rules for determining definite assignment	81
10.4.4.1 General	81
10.4.4.2 General rules for statements	82
10.4.4.3 Block statements, checked, and unchecked statements	82
10.4.4.4 Expression statements	82
10.4.4.5 Declaration statements	83
10.4.4.6 If statements	83
10.4.4.7 Switch statements	83
10.4.4.8 While statements	83
10.4.4.9 Do statements	83

10.4.4.10 For statements	84
10.4.4.11 Break, continue, and goto statements.....	84
10.4.4.12 Throw statements	84
10.4.4.13 Return statements	84
10.4.4.14 Try-catch statements	85
10.4.4.15 Try-finally statements	85
10.4.4.16 Try-catch-finally statements	85
10.4.4.17 Foreach statements	86
10.4.4.18 Using statements	86
10.4.4.19 Lock statements	87
10.4.4.20 Yield statements.....	87
10.4.4.21 General rules for constant expressions	87
10.4.4.22 General rules for simple expressions	88
10.4.4.23 General rules for expressions with embedded expressions	88
10.4.4.24 Invocation expressions and object creation expressions	88
10.4.4.25 Simple assignment expressions	89
10.4.4.26 && expressions	89
10.4.4.27 expressions	90
10.4.4.28 ! expressions	91
10.4.4.29 ?? expressions	91
10.4.4.30 ?: expressions.....	91
10.4.4.31 Anonymous functions	92
10.5 Variable references	92
10.6 Atomicity of variable references.....	92
11. Conversions	93
11.1 General	93
11.2 Implicit conversions.....	93
11.2.1 General.....	93
11.2.2 Identity conversion	94
11.2.3 Implicit numeric conversions	94
11.2.4 Implicit enumeration conversions	94
11.2.5 Implicit nullable conversions.....	94
11.2.6 Null literal conversions.....	94
11.2.7 Implicit reference conversions.....	95
11.2.8 Boxing conversions.....	95
11.2.9 Implicit dynamic conversions.....	97
11.2.10 Implicit constant expression conversions	97
11.2.11 Implicit conversions involving type parameters	97
11.2.12 User-defined implicit conversions	98
11.2.13 Anonymous function conversions and method group conversions	98
11.3 Explicit conversions	98
11.3.1 General.....	98
11.3.2 Explicit numeric conversions.....	99
11.3.3 Explicit enumeration conversions.....	101
11.3.4 Explicit nullable conversions	101
11.3.5 Explicit reference conversions	101
11.3.6 Unboxing conversions	102
11.3.7 Explicit dynamic conversions	103
11.3.8 Explicit conversions involving type parameters.....	103

11.3.9 User-defined explicit conversions.....	105
11.4 Standard conversions.....	105
11.4.1 General.....	105
11.4.2 Standard implicit conversions.....	105
11.4.3 Standard explicit conversions.....	105
11.5 User-defined conversions.....	105
11.5.1 General.....	105
11.5.2 Permitted user-defined conversions.....	105
11.5.3 Evaluation of user-defined conversions.....	106
11.5.4 User-defined implicit conversions.....	107
11.5.5 User-defined explicit conversions.....	107
11.6 Conversions involving nullable types.....	109
11.6.1 Nullable Conversions.....	109
11.6.2 Lifted conversions.....	109
11.7 Anonymous function conversions.....	109
11.7.1 General.....	109
11.7.2 Evaluation of anonymous function conversions to delegate types.....	111
11.7.3 Evaluation of anonymous function conversions to expression tree types.....	112
11.8 Method group conversions.....	112
12. Expressions.....	115
12.1 General.....	115
12.2 Expression classifications.....	115
12.2.1 General.....	115
12.2.2 Values of expressions.....	116
12.3 Static and Dynamic Binding.....	116
12.3.1 General.....	116
12.3.2 Binding-time.....	117
12.3.3 Dynamic binding.....	117
12.3.4 Types of subexpressions.....	118
12.4 Operators.....	118
12.4.1 General.....	118
12.4.2 Operator precedence and associativity.....	118
12.4.3 Operator overloading.....	119
12.4.4 Unary operator overload resolution.....	121
12.4.5 Binary operator overload resolution.....	121
12.4.6 Candidate user-defined operators.....	121
12.4.7 Numeric promotions.....	122
12.4.7.1 General.....	122
12.4.7.2 Unary numeric promotions.....	122
12.4.7.3 Binary numeric promotions.....	122
12.4.8 Lifted operators.....	123
12.5 Member lookup.....	124
12.5.1 General.....	124
12.5.2 Base types.....	125
12.6 Function members.....	126
12.6.1 General.....	126
12.6.2 Argument lists.....	128
12.6.2.1 General.....	128
12.6.2.2 Corresponding parameters.....	129