Microsoft Partner
Silver Learning

C# Стартовый

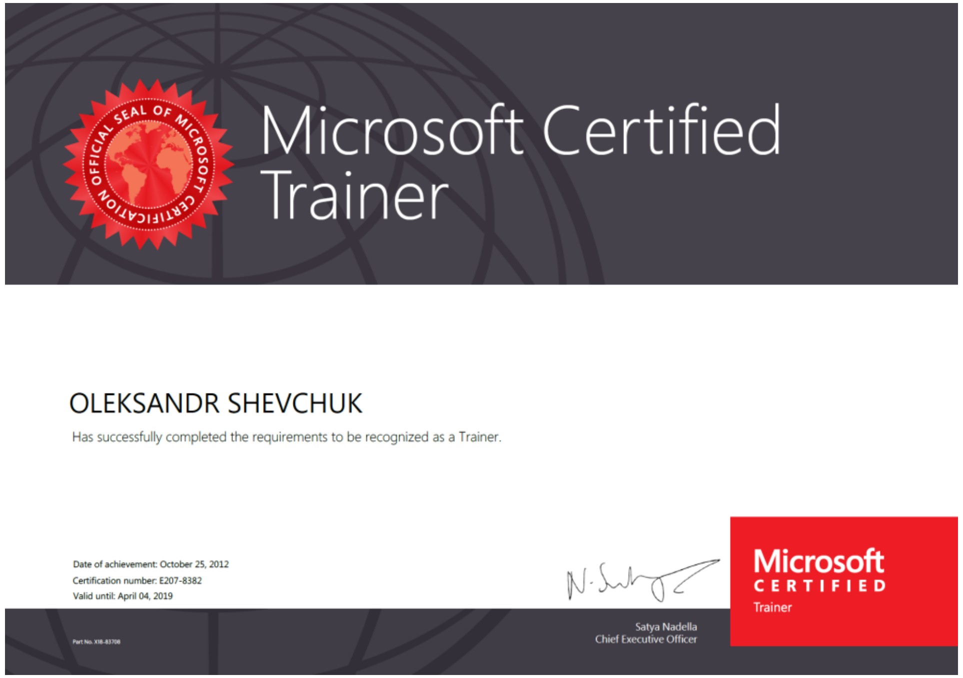
ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Арифметические операторы

ITVDN
IT VIDEO DEVELOPERS NETWORK

Introduction

Александр Шевчук



OFFICIAL SEAL OF MICROSOFT CERTIFICATION

Microsoft Certified Trainer

OLEKSANDR SHEVCHUK

Has successfully completed the requirements to be recognized as a Trainer.

Date of achievement: October 25, 2012
Certification number: E207-8382
Valid until: April 04, 2019

Part No. X18-83708

Satya Nadella
Chief Executive Officer

Microsoft CERTIFIED Trainer

MCID: 9230440

Тема урока

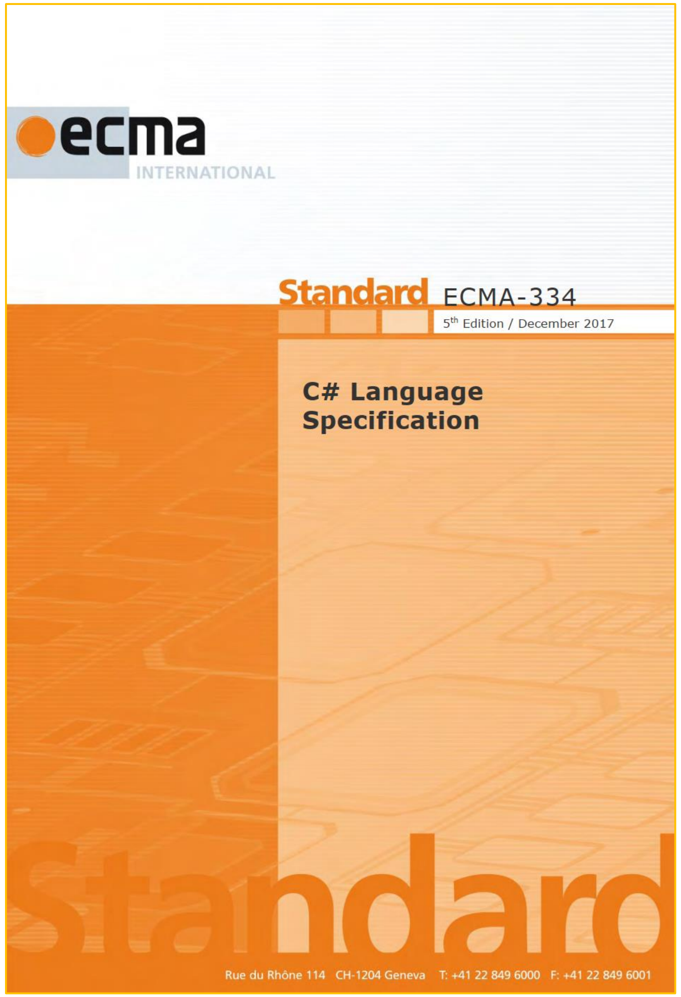
Арифметические операторы

Операторы арифметических действий над двумя операндами

```
// Сложение
int int.operator + (int left, int right)
uint uint.operator + (uint left, uint right)
long long.operator + (long left, long right)
ulong ulong.operator + (ulong left, ulong right)
float float.operator + (float left, float right)
double double.operator + (double left, double right)
decimal decimal.operator + (decimal left, decimal right)
// Вычитание
int int.operator - (int left, int right)
uint uint.operator - (uint left, uint right)
long long.operator - (long left, long right)
ulong ulong.operator - (ulong left, ulong right)
float float.operator - (float left, float right)
double double.operator - (double left, double right)
decimal decimal.operator - (decimal left, decimal right)
// Умножение
int int.operator * (int left, int right)
uint uint.operator * (uint left, uint right)
long long.operator * (long left, long right)
ulong ulong.operator * (ulong left, ulong right)
float float.operator * (float left, float right)
double double.operator * (double left, double right)
decimal decimal.operator * (decimal left, decimal right)
```

```
// Деление нацело
int int.operator / (int left, int right)
uint uint.operator / (uint left, uint right)
long long.operator / (long left, long right)
ulong ulong.operator / (ulong left, ulong right)
float float.operator / (float left, float right)
double double.operator / (double left, double right)
decimal decimal.operator / (decimal left, decimal right)
// Деление с получением остатка
int int.operator % (int left, int right)
uint uint.operator % (uint left, uint right)
long long.operator % (long left, long right)
ulong ulong.operator % (ulong left, ulong right)
float float.operator % (float left, float right)
double double.operator % (double left, double right)
decimal decimal.operator % (decimal left, decimal right)
```

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#



ecma INTERNATIONAL

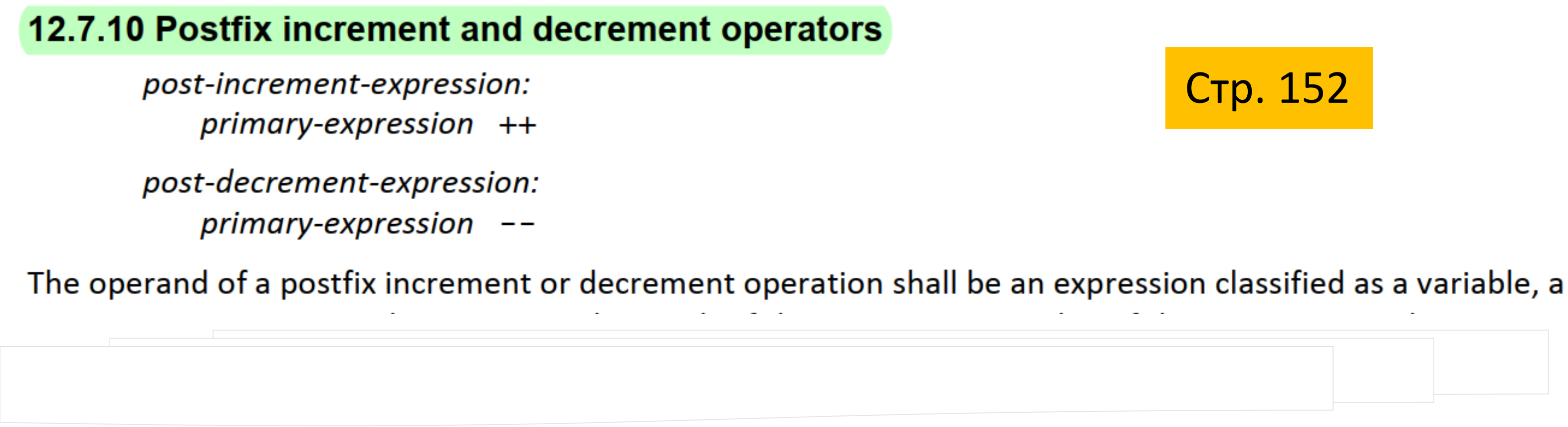
Standard ECMA-334

5th Edition / December 2017

C# Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001



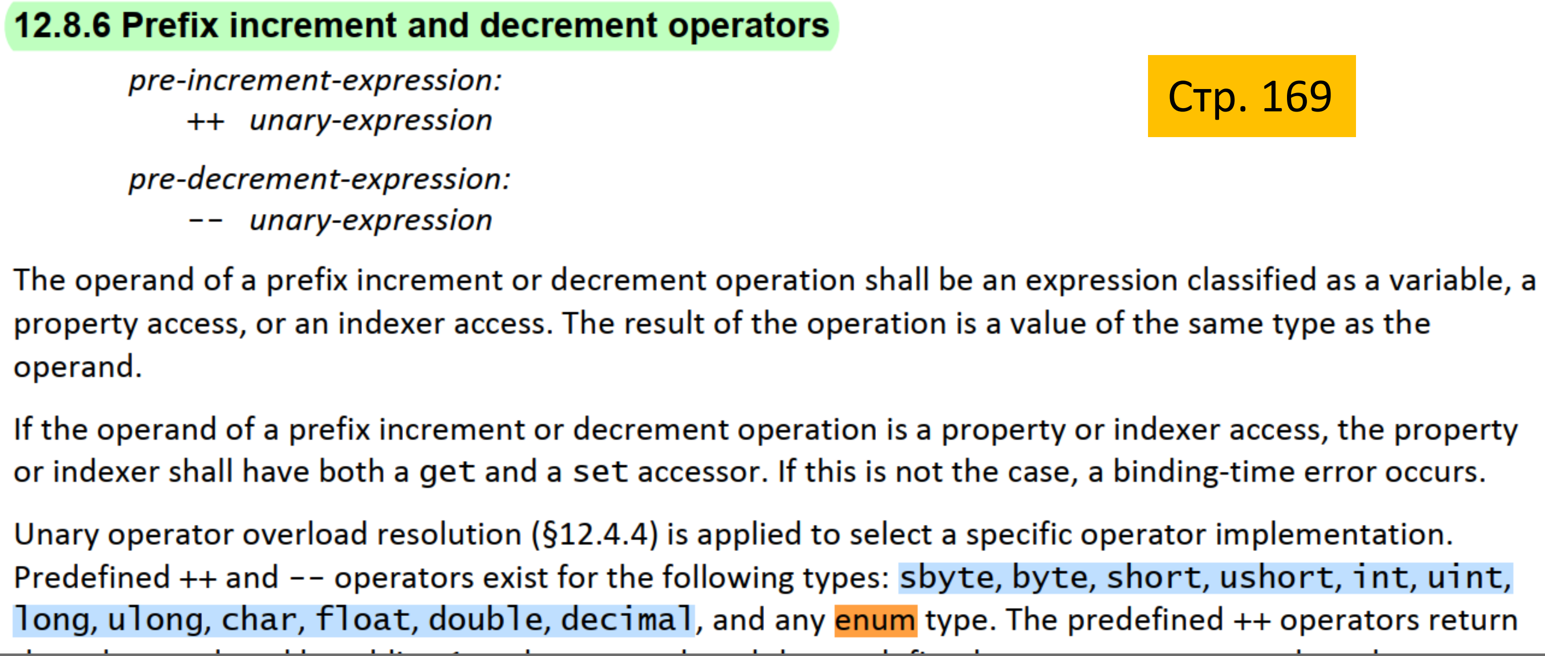
12.7.10 Postfix increment and decrement operators

post-increment-expression:
    primary-expression ++

post-decrement-expression:
    primary-expression --

The operand of a postfix increment or decrement operation shall be an expression classified as a variable, a

Стр. 152



12.8.6 Prefix increment and decrement operators

pre-increment-expression:
    ++ *unary-expression*

pre-decrement-expression:
    -- *unary-expression*

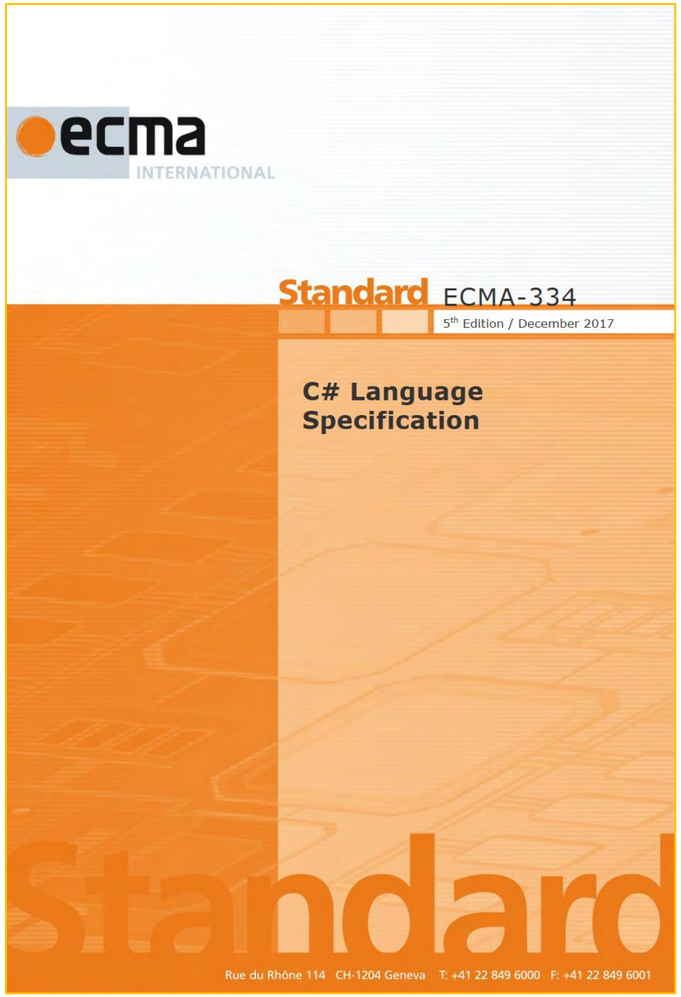
The operand of a prefix increment or decrement operation shall be an expression classified as a variable, a property access, or an indexer access. The result of the operation is a value of the same type as the operand.

If the operand of a prefix increment or decrement operation is a property or indexer access, the property or indexer shall have both a `get` and a `set` accessor. If this is not the case, a binding-time error occurs.

Unary operator overload resolution (§12.4.4) is applied to select a specific operator implementation. Predefined ++ and -- operators exist for the following types: `sbyte`, `byte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`, `char`, `float`, `double`, `decimal`, and any enum type. The predefined ++ operators return

Стр. 169

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#



ecma INTERNATIONAL

Standard ECMA-334

5th Edition / December 2017

C# Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

12.9 Arithmetic operators

Стр. 172

12.9.1 General

The *, /, %, +, and – operators are called the arithmetic operators.

● ● ●

12.9.2 Multiplication operator

Стр. 172

For an operation of the form x * y, binary operator overload resolution (§12.4.5) is applied to select a specific operator implementation. The operands are converted to the parameter types of the selected

● ● ●

12.9.3 Division operator

Стр. 173

For an operation of the form x / y, binary operator overload resolution (§12.4.5) is applied to select a specific operator implementation. The operands are converted to the parameter types of the selected

● ● ●

12.9.4 Remainder operator

Стр. 174

For an operation of the form x % y, binary operator overload resolution (§12.4.5) is applied to select a specific operator implementation. The operands are converted to the parameter types of the selected

● ● ●

12.9.5 Addition operator

Стр. 175

For an operation of the form x + y, binary operator overload resolution (§12.4.5) is applied to select a specific operator implementation. The operands are converted to the parameter types of the selected

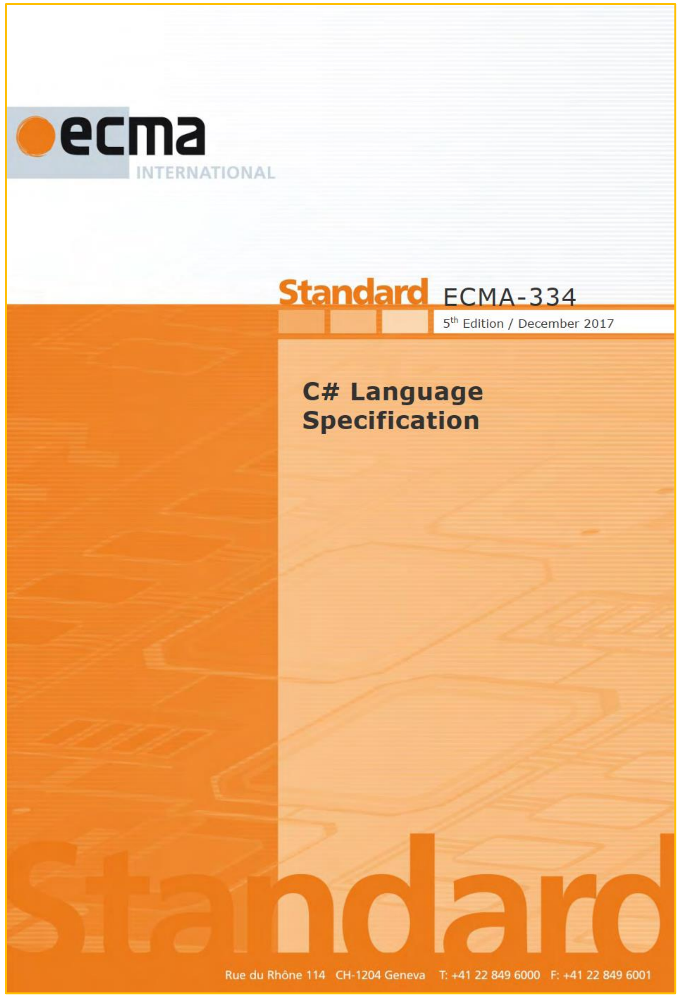
● ● ●

12.9.6 Subtraction operator

Стр. 176

For an operation of the form x – y, binary operator overload resolution (§12.4.5) is applied to select a specific operator implementation. The operands are converted to the parameter types of the selected

● ● ●



ecma INTERNATIONAL

Standard ECMA-334

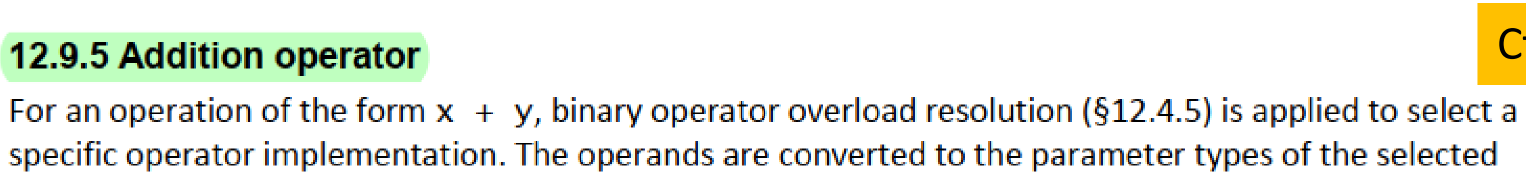
5th Edition / December 2017

C# Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

Стр. 175

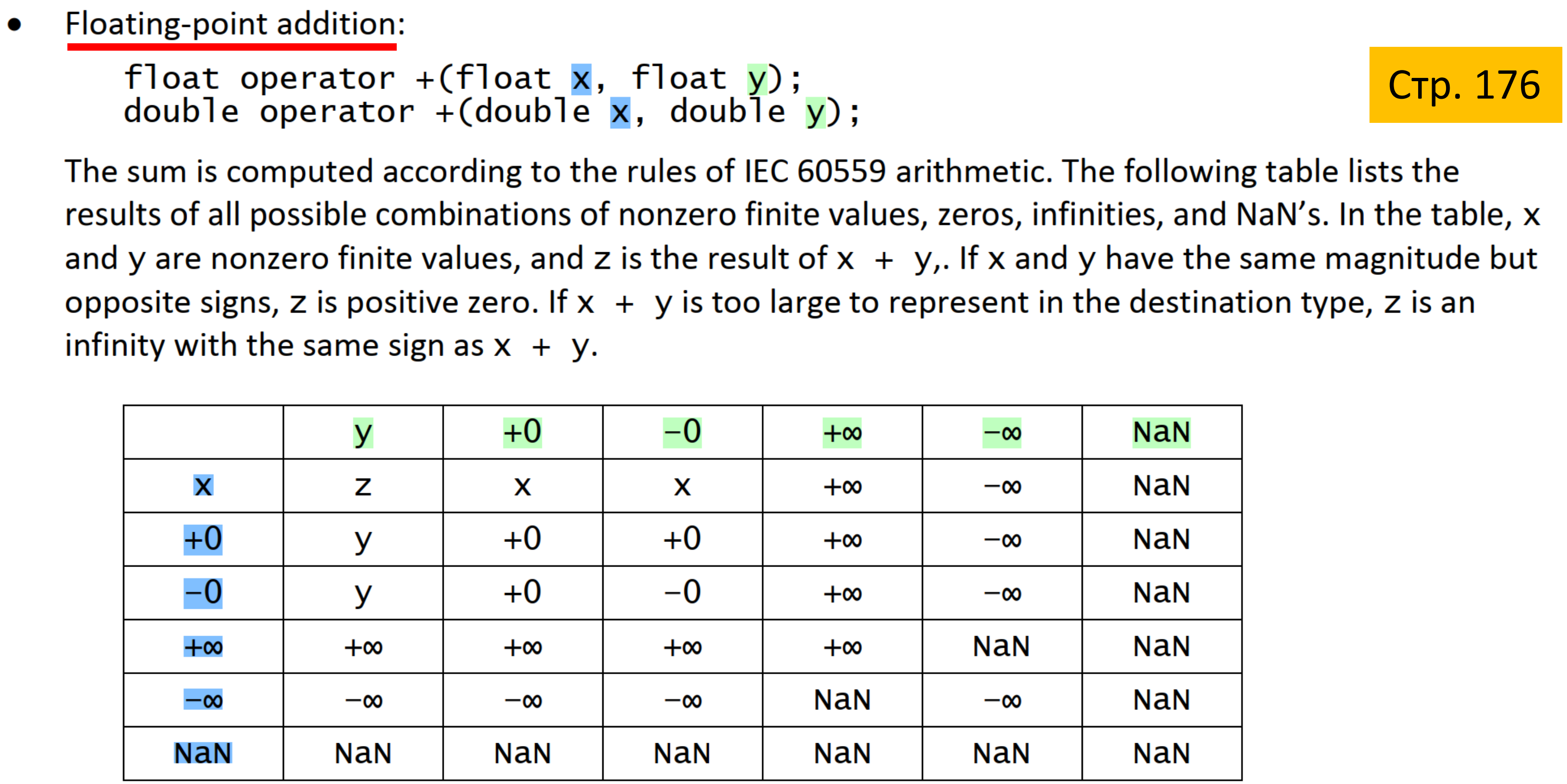


12.9.5 Addition operator

For an operation of the form x + y, binary operator overload resolution (§12.4.5) is applied to select a specific operator implementation. The operands are converted to the parameter types of the selected

• • •

Стр. 176



- Floating-point addition:

```
float operator +(float x, float y);
double operator +(double x, double y);
```

The sum is computed according to the rules of IEC 60559 arithmetic. The following table lists the results of all possible combinations of nonzero finite values, zeros, infinities, and NaN’s. In the table, x and y are nonzero finite values, and z is the result of x + y,. If x and y have the same magnitude but opposite signs, z is positive zero. If x + y is too large to represent in the destination type, z is an infinity with the same sign as x + y.

	y	+0	–0	+∞	–∞	NaN
x	z	x	x	+∞	–∞	NaN
+0	y	+0	+0	+∞	–∞	NaN
–0	y	+0	–0	+∞	–∞	NaN
+∞	+∞	+∞	+∞	+∞	NaN	NaN
–∞	–∞	–∞	–∞	NaN	–∞	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN

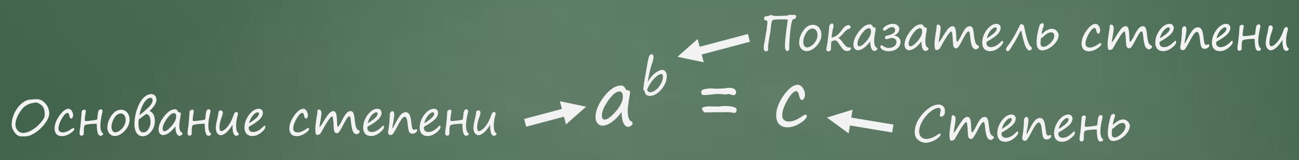
• • •

Степень. Корень. Логарифм.

Показатель степени

Основание степени → $a^b = c$ ← Степень

Степень. Корень. Логарифм.



Показатель степени

Основание степени → $a^b = c$ ← Степень

1) Нахождение степени:

$c = ?$ $6^2 = c;$

Степень. Корень. Логарифм.

Показатель степени

Основание степени → $a^b = c$ ← Степень

1) Нахождение степени:

$c = ?$ $6^2 = c;$ $6 \times 6 = 36;$ $c = 36$

Степень. Корень. Логарифм.

Показатель степени

Основание степени → $a^b = c$ ← Степень

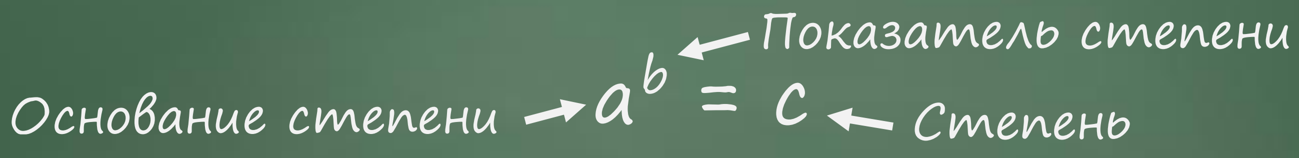
1) Нахождение степени:

$c = ?$ $6^2 = c$; $6 \times 6 = 36$; $c = 36$

2) Нахождение основания степени:

$a = ?$ $a^2 = 36$;

Степень. Корень. Логарифм.



Показатель степени

Основание степени → $a^b = c$ ← Степень

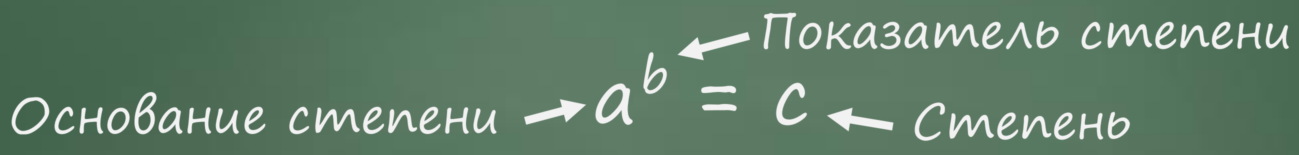
1) Нахождение степени:

$c = ?$ $6^2 = c;$ $6 \times 6 = 36;$ $c = 36$

2) Нахождение основания степени:

$a = ?$ $a^2 = 36;$ $a = \sqrt{36};$ $a = 6$

Степень. Корень. Логарифм.



1) Нахождение степени:

$c = ?$ $6^2 = c;$ $6 \times 6 = 36;$ $c = 36$

2) Нахождение основания степени:

$a = ?$ $a^2 = 36;$ $a = \sqrt{36};$ $a = 6$

3) Нахождение показателя степени:

$b = ?$ $6^b = 36;$

Степень. Корень. Логарифм.

Показатель степени

Основание степени → $a^b = c$ ← Степень

1) Нахождение степени:

$c = ?$ $6^2 = c;$ $6 \times 6 = 36;$ $c = 36$

2) Нахождение основания степени:

$a = ?$ $a^2 = 36;$ $a = \sqrt{36};$ $a = 6$

3) Нахождение показателя степени:

$b = ?$ $6^b = 36;$ $b = \log_6 36;$ $b = 2$

Степень. Корень. Логарифм.

Показатель степени

Основание степени → $a^b = c$ ← Степень

1) Нахождение степени: c = Math.Pow(6, 2);

$c = ?$ $6^2 = c;$ $6 \times 6 = 36;$ $c = 36$

2) Нахождение основания степени: a = Math.Sqrt(36);

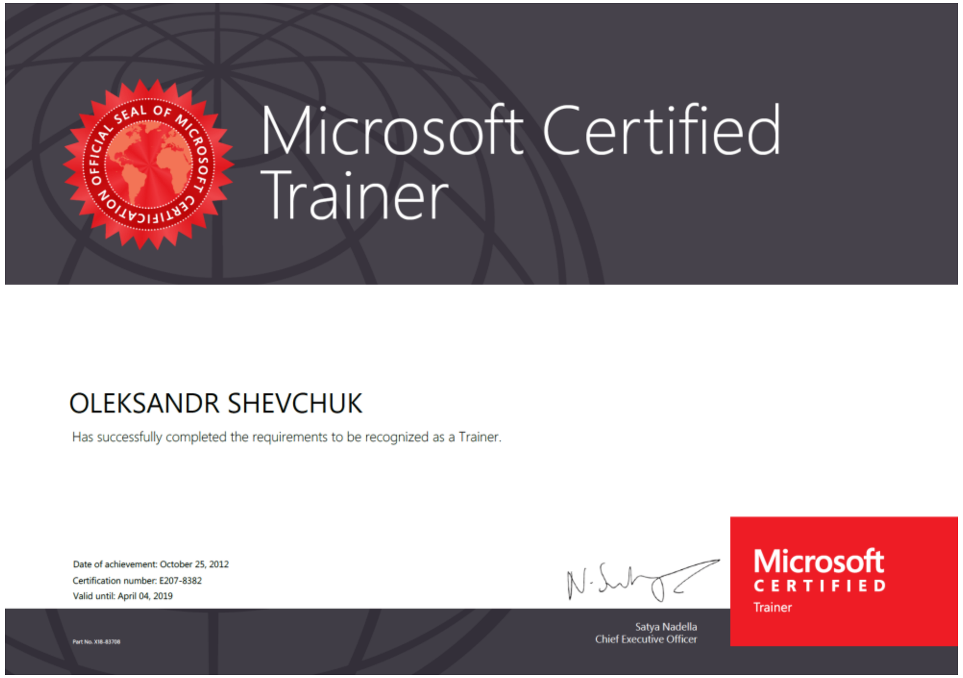
$a = ?$ $a^2 = 36;$ $a = \sqrt{36};$ $a = 6$

3) Нахождение показателя степени: b = Math.Log(36, 6);

$b = ?$ $6^b = 36;$ $b = \log_6 36;$ $b = 2$

Спасибо за внимание! До новых встреч!

Александр Шевчук



Microsoft Certified Trainer

OFFICIAL SEAL OF MICROSOFT CERTIFICATION

OLEKSANDR SHEVCHUK

Has successfully completed the requirements to be recognized as a Trainer.

Date of achievement: October 25, 2012
Certification number: E207-8382
Valid until: April 04, 2019

Satya Nadella
Chief Executive Officer

Microsoft CERTIFIED Trainer

Part No. X18-83708

MCID: 9230440

Информационный видеоресурс для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK