Microsoft Partner
Silver Learning

C# Стартовый

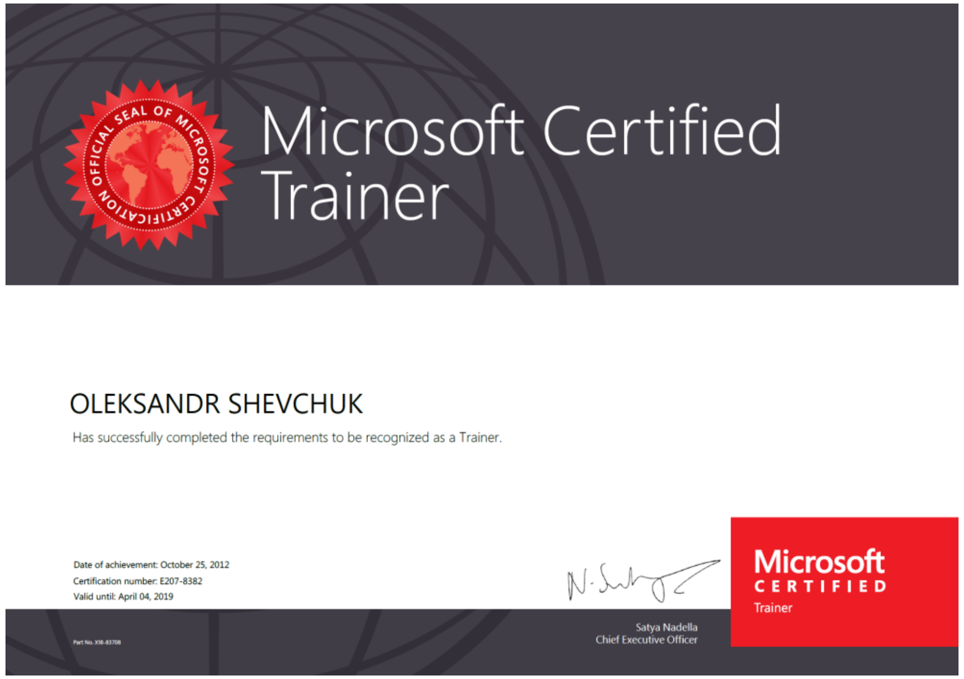
ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Операции над строковыми переменными

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

[illegible]
Satya Nadella
Chief Executive Officer

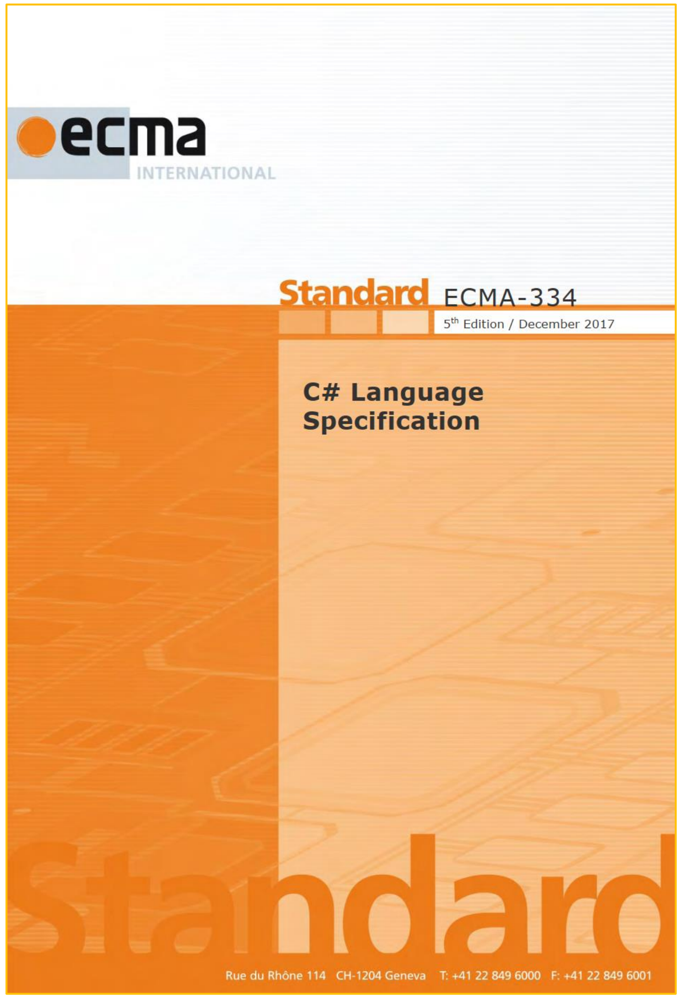
Microsoft CERTIFIED Trainer

MCID: 9230440

Тема урока

Операции над строковыми переменными

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

12.9.5 Addition operator

Стр. 175

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

The predefined addition operators are listed below. For numeric and enumeration types, the predefined addition operators compute the sum of the two operands. When one or both operands are of type **`string`**, the predefined addition operators concatenate the string representation of the operands.

```
class Test
{
    static void Main() {
        string s = null;
        Console.WriteLine("s = >" + s + "<"); // displays s = ><
        int i = 1;
        Console.WriteLine("i = " + i);        // displays i = 1
        float f = 1.2300E+15F;
        Console.WriteLine("f = " + f);        // displays f = 1.23E+15
        decimal d = 2.900m;
        Console.WriteLine("d = " + d);        // displays d = 2.900
    }
}
```

Стр. 177

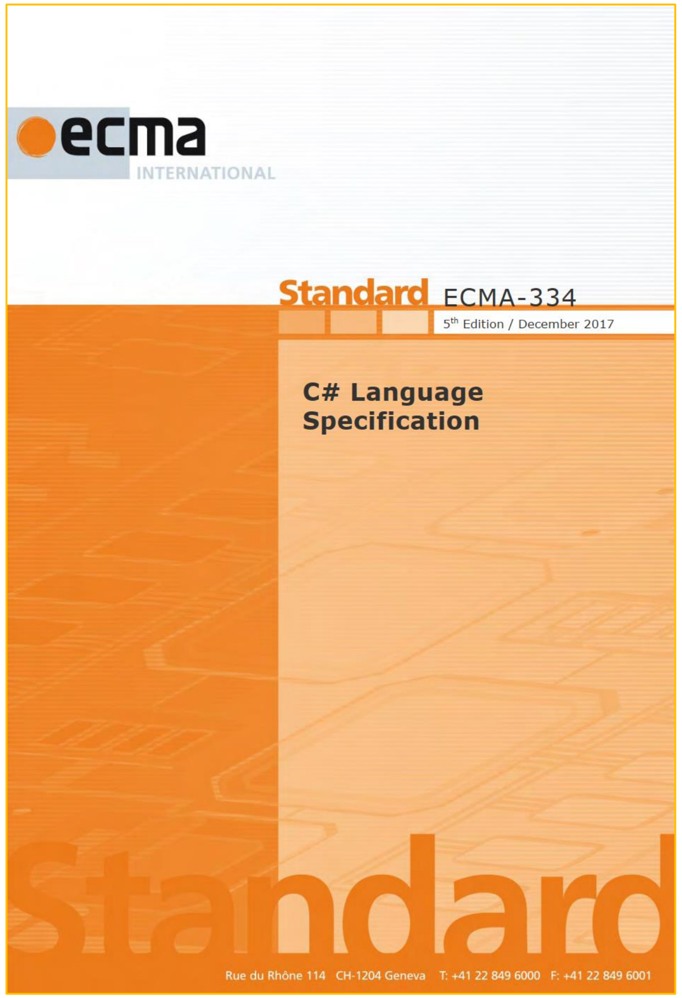
The output shown in the comments is the typical result on a US-English system. The precise output might depend on the regional settings of the execution environment. The string-concatenation operator itself behaves the same way in each case, but the **`ToString`** methods implicitly called during execution might be affected by regional settings. *end example*]

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Введение промежуточной переменной
Присвойте результат вычисления выражения промежуточной переменной, имя которой резюмирует суть выражения.

Стр. 559

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#


ecma INTERNATIONAL

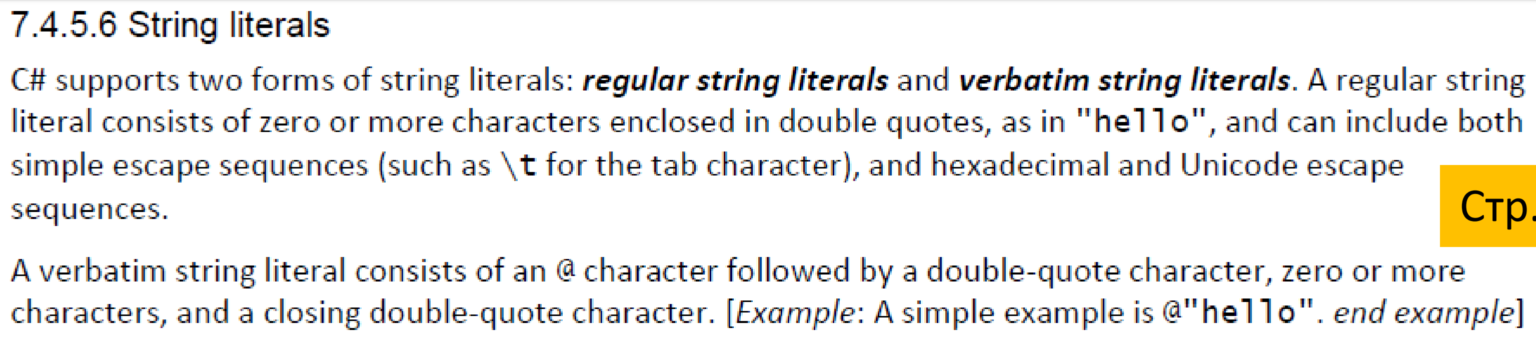
Standard ECMA-334

5th Edition / December 2017

C# Language Specification

Standard

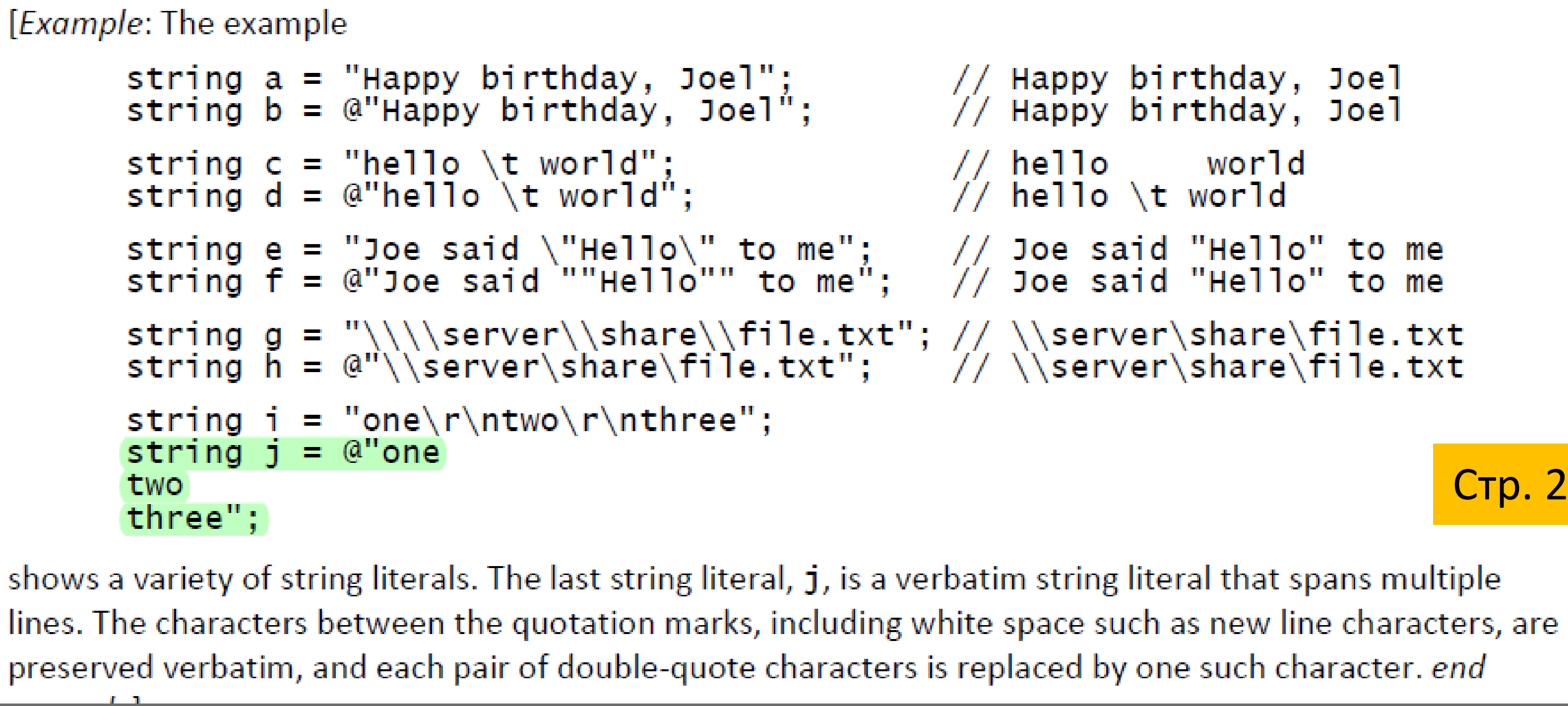
Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001


7.4.5.6 String literals

C# supports two forms of string literals: ***regular string literals*** and ***verbatim string literals***. A regular string literal consists of zero or more characters enclosed in double quotes, as in `"hello"`, and can include both simple escape sequences (such as `\t` for the tab character), and hexadecimal and Unicode escape sequences.

A verbatim string literal consists of an `@` character followed by a double-quote character, zero or more characters, and a closing double-quote character. [*Example*: A simple example is `@"hello"`. *end example*]

Стр. 25

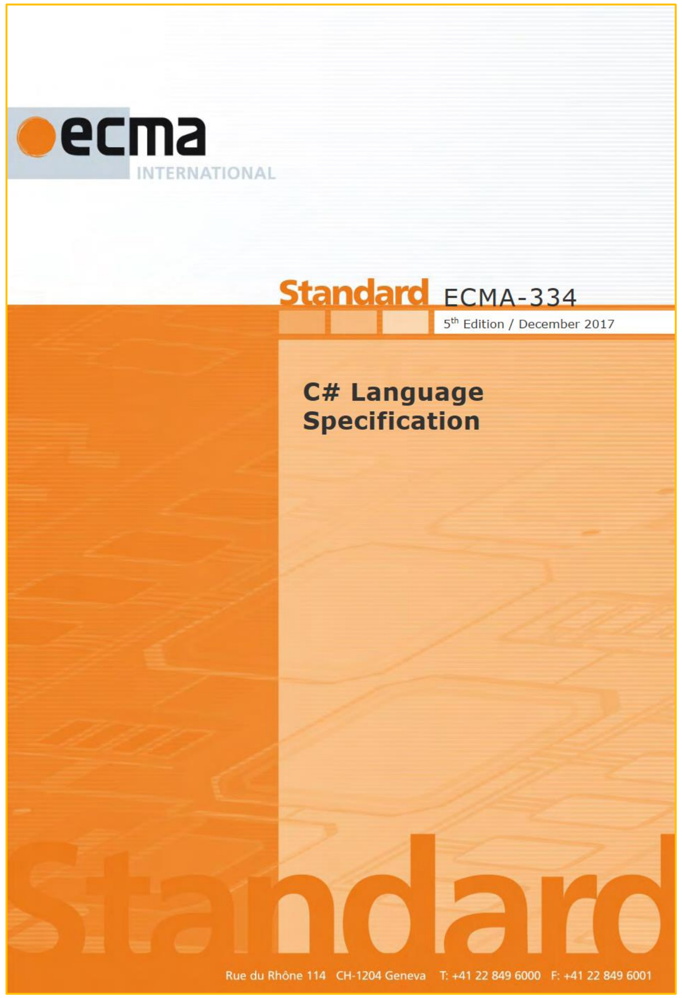

[*Example*: The example

```
string a = "Happy birthday, Joel";        // Happy birthday, Joel
string b = @"Happy birthday, Joel";       // Happy birthday, Joel

string c = "hello \t world";              // hello     world
string d = @"hello \t world";             // hello \t world

string e = "Joe said \"Hello\" to me";    // Joe said "Hello" to me
string f = @"Joe said ""Hello"" to me";   // Joe said "Hello" to me

string g = "\\\\server\\share\\file.txt"; // \\server\share\file.txt
string h = @"\\server\share\file.txt";    // \\server\share\file.txt

string i = "one\r\ntwo\r\nthree";
string j = @"one
two
three";
```

Стр. 26

shows a variety of string literals. The last string literal, `j`, is a verbatim string literal that spans multiple lines. The characters between the quotation marks, including white space such as new line characters, are preserved verbatim, and each pair of double-quote characters is replaced by one such character. *end*

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

7.4.5.6 String literals

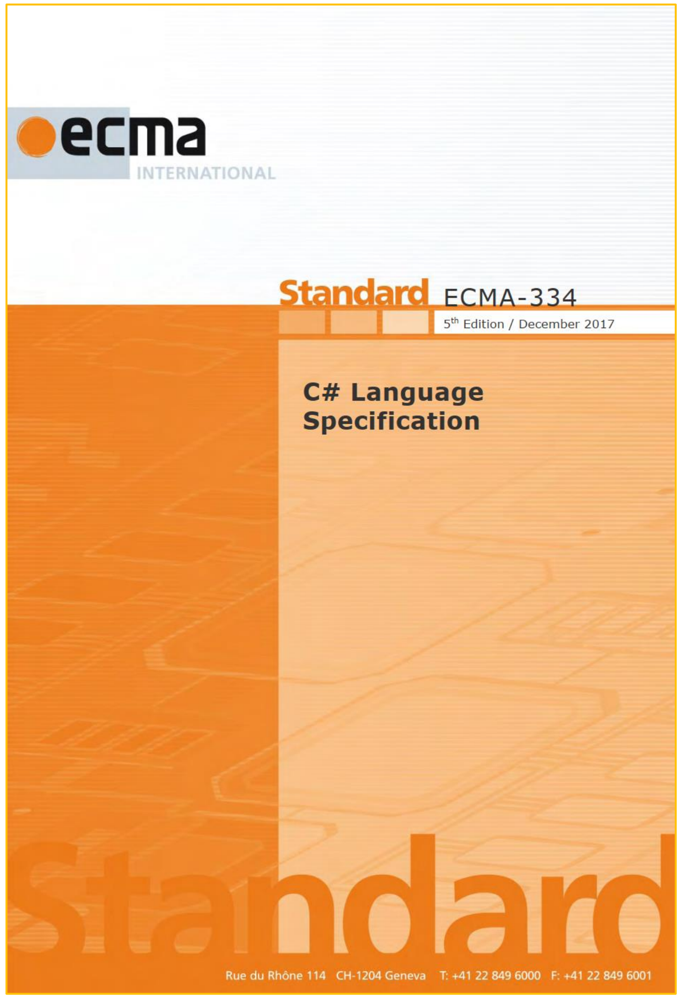
C# supports two forms of string literals: ***regular string literals*** and ***verbatim string literals***. A regular string literal consists of zero or more characters enclosed in double quotes, as in `"hello"`, and can include both simple escape sequences (such as `\t` for the tab character), and hexadecimal and Unicode escape sequences.

Стр. 25

Стр. 26

```
string a = "Happy birthday, Joel";        // Happy birthday, Joel
string b = @"Happy birthday, Joel";       // Happy birthday, Joel

string c = "hello \t world";              // hello     world
string d = @"hello \t world";             // hello \t world

string e = "Joe said \"Hello\" to me";    // Joe said "Hello" to me
string f = @"Joe said ""Hello"" to me";   // Joe said "Hello" to me

string g = "\\\\server\\share\\file.txt"; // \\server\share\file.txt
string h = @"\\server\share\file.txt";    // \\server\share\file.txt

string i = "one\r\ntwo\r\nthree";
string j = @"one
two
three";
```

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#



7.4.5.5 Character literals

A character literal represents a single character, and consists of a character in quotes, as in `'a'`.

Стр. 24

A simple escape sequence represents a Unicode character, as described in the table below.

Стр. 25

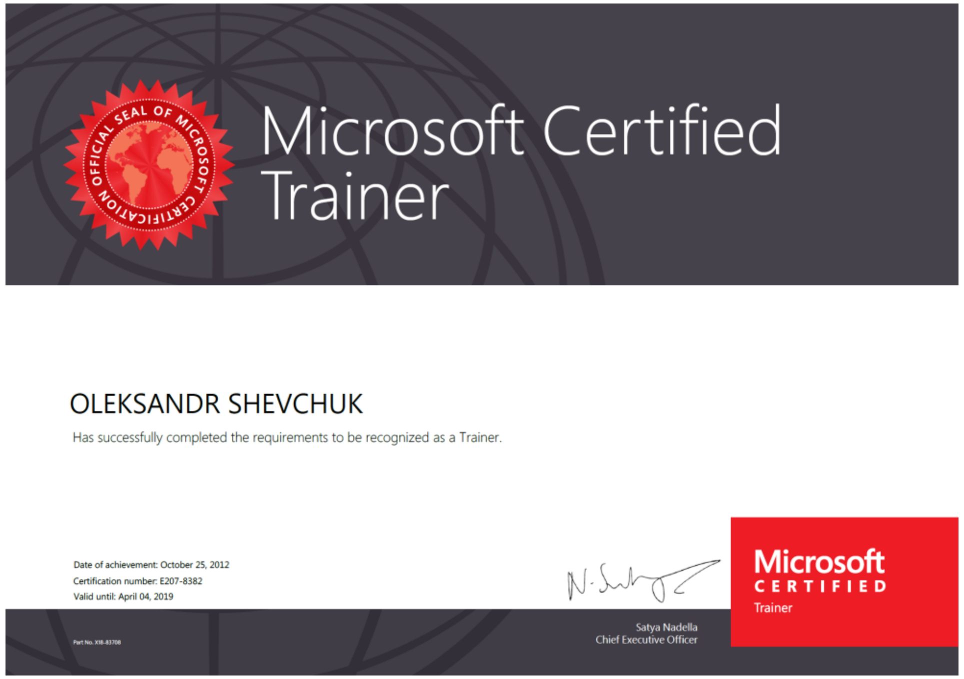
Escape sequence	Character name	Unicode code point
\'	Single quote	U+0027
\"	Double quote	U+0022
\\	Backslash	U+005C
\0	Null	U+0000
\a	Alert	U+0007
\b	Backspace	U+0008
\f	Form feed	U+000C
\n	New line	U+000A
\r	Carriage return	U+000D
\t	Horizontal tab	U+0009
\v	Vertical tab	U+000B

\x - hexadecimal-escape-sequence

\u - unicode-escape-sequence

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

Part No. X18-83708

[illegible]

Satya Nadella
Chief Executive Officer

Microsoft CERTIFIED Trainer

MCID: 9230440

Информационный видеоресурс для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK