



Microsoft Partner
Silver Learning

C# Стартовый

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Логические величины и операции сравнения



ITVDN
IT VIDEO DEVELOPERS NETWORK

Introduction



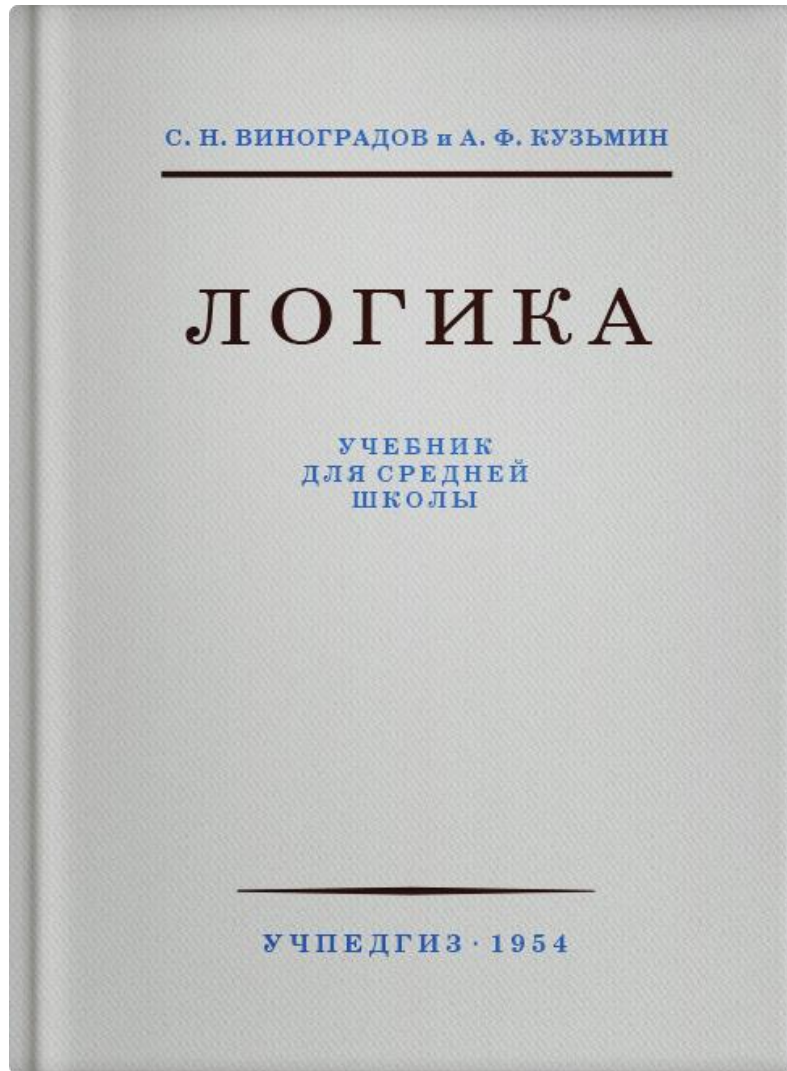
Александр Шевчук



MCID: 9230440

Тема урока

Логические величины и операции сравнения



Логика — это наука изучающая высказывания



Высказывание — предложение, выражающее суждение.



Суждение — форма мышления, в которой что-либо утверждается или отрицается о предмете, его свойствах или отношениях между предметами.

Высказывание

Высказывание – повествовательное предложение. Высказывание может быть истинным или ложным. Высказывание может быть выражено с помощью слов, математических, физических и прочих формул.

Примеры высказываний:

Марс дальше от Солнца чем Венера. (**истинное высказывание**)

$2 + 6 > 8$ (**ложное высказывание**)

$II + VI > VIII$ (**ложное высказывание**)

Сумма чисел 2 и 6 больше числа 8. (**ложное высказывание**)

Эти три высказывания
означают одно и то же, но
выражены разными словами

В нашей галактике существуют внеземные цивилизации. (высказывание либо **истинно** либо **ложно**)

Не всякое предложение является высказыванием

Восклицательные и вопросительные предложения и определения высказываниями не являются.

Примеры не высказываний:

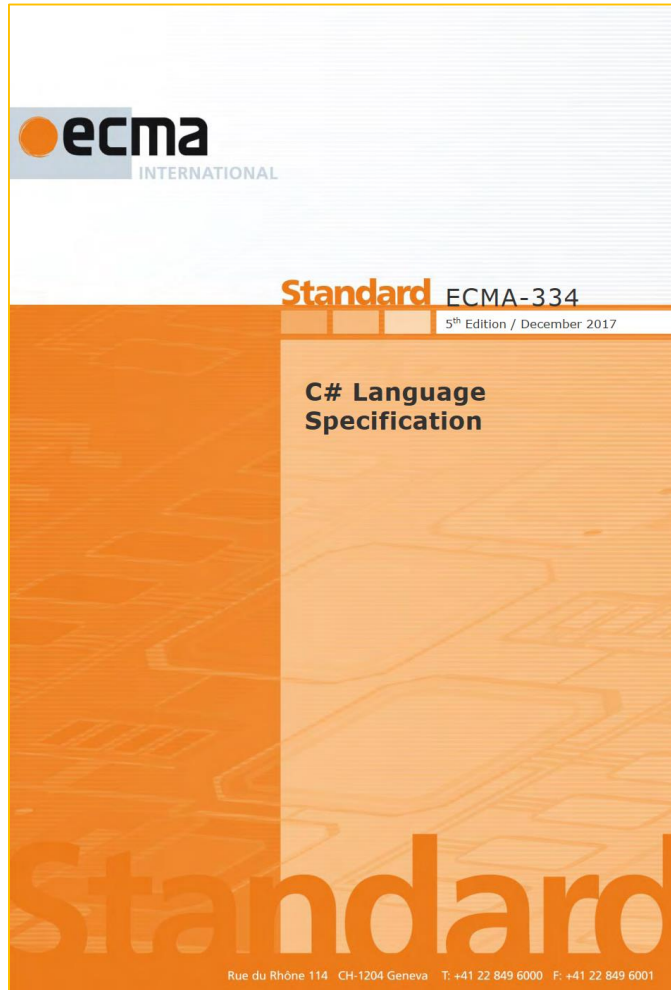
- ? Сколько тебе лет? Как тебя зовут? Какой у тебя компьютер? Чем ты интересуешься? Сколько время?
- ! Стоп! Пейте гранатовый сок! Положи книгу на место! Не шуметь!

Определение

Назовем медианой отрезок, соединяющий вершину треугольника с серединой противоположной стороны.



В определении лишь устанавливается название некоторого объекта. Это название можно дать и другому объекту а этому объекту дать другое название. Поэтому определения не могут быть истинными или ложными.



12.11 Relational and type-testing operators

Стр. 180

12.11.1 General

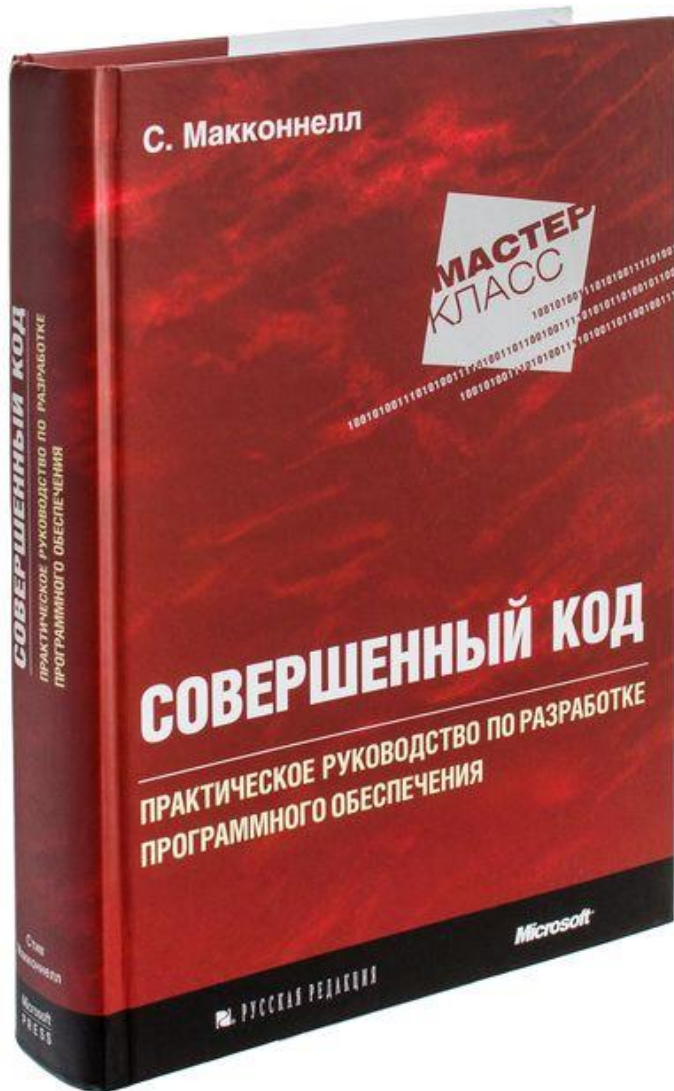
The `==`, `!=`, `<`, `>`, `<=`, `>=`, `is`, and `as` operators are called the relational and type-testing operators.



The predefined comparison operators are described in the following subclauses. All predefined comparison operators return a result of type `bool`, as described in the following table.

	Operation	Result
1	<code>x == y</code>	<code>true</code> if <code>x</code> is equal to <code>y</code> , <code>false</code> otherwise
2	<code>x != y</code>	<code>true</code> if <code>x</code> is not equal to <code>y</code> , <code>false</code> otherwise
3	<code>x < y</code>	<code>true</code> if <code>x</code> is less than <code>y</code> , <code>false</code> otherwise
4	<code>x > y</code>	<code>true</code> if <code>x</code> is greater than <code>y</code> , <code>false</code> otherwise
5	<code>x <= y</code>	<code>true</code> if <code>x</code> is less than or equal to <code>y</code> , <code>false</code> otherwise
6	<code>x >= y</code>	<code>true</code> if <code>x</code> is greater than or equal to <code>y</code> , <code>false</code> otherwise

Стр. 181



Именованние булевых переменных

...

Некоторые программисты любят дополнять имена булевых переменных префиксом `is`. В результате имя переменной превращается в вопрос: `isdone?` `isError?` `isFound?` `isProcessingComplete?` Ответ на этот вопрос сразу становится и значением переменной.



Достоинство этого подхода в том, что он исключает использование неопределенных имен: вопрос `isStatus?` не имеет никакого смысла.



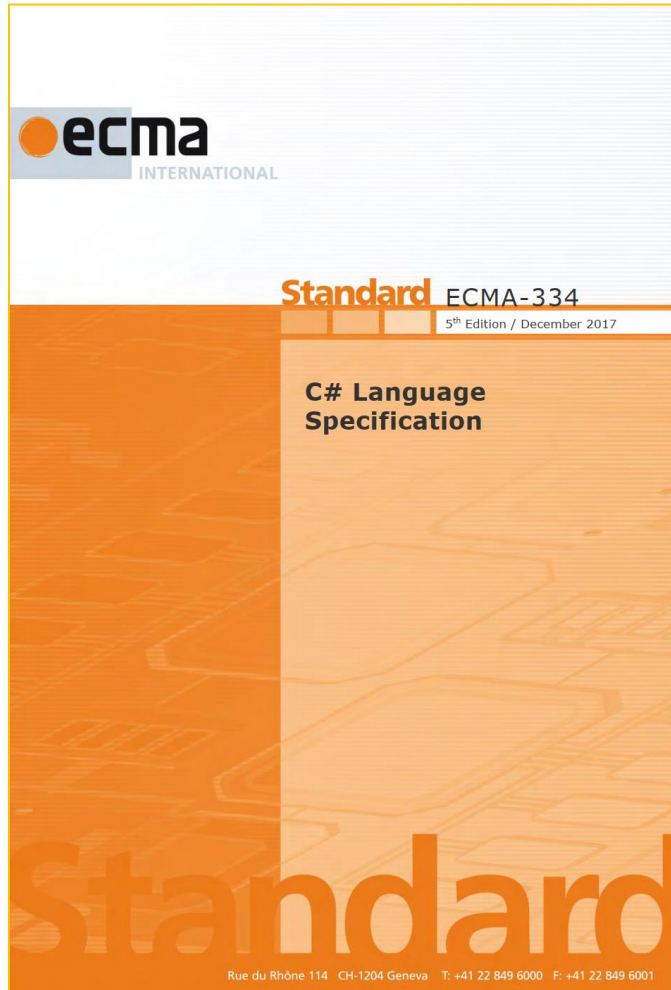
Однако в то же время он затрудняет чтение логических выражений: например, условие `if(isFound)` менее понятно, чем `if(found)`.

...

ГЛАВА 11. Сила имен переменных.

Главка: «Именованние булевых переменных»

Стр. 261



12.11.3 Floating-point comparison operators

The predefined floating-point comparison operators are:

```
bool operator ==(float x, float y);
bool operator ==(double x, double y);

bool operator !=(float x, float y);
bool operator !=(double x, double y);

bool operator <(float x, float y);
bool operator <(double x, double y);

bool operator >(float x, float y);
bool operator >(double x, double y);

bool operator <=(float x, float y);
bool operator <=(double x, double y);

bool operator >=(float x, float y);
bool operator >=(double x, double y);
```

The operators compare the operands according to the rules of the IEC 60559 standard:

If either operand is NaN, the result is **false** for all operators except **!=**, for which the result is **true**. For any two operands, $x \neq y$ always produces the same result as $!(x == y)$. However, when one or both operands are NaN, the **<**, **>**, **<=**, and **>=** operators do *not* produce the same results as the logical negation of the opposite operator. [Example: If either of x and y is NaN, then $x < y$ is **false**, but $!(x >= y)$ is **true**. end example]

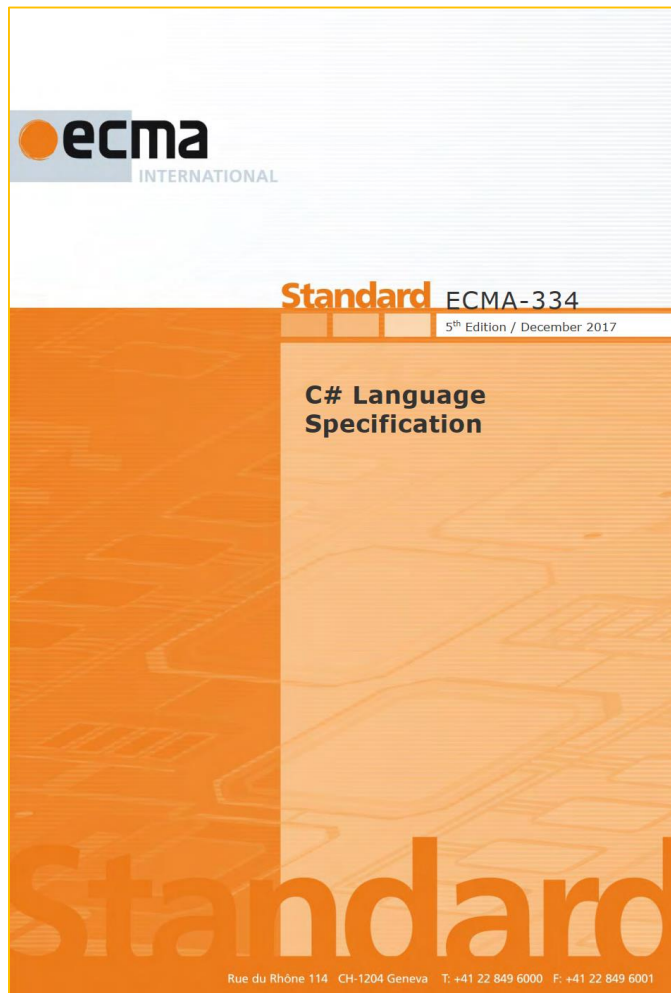
- When neither operand is NaN, the operators compare the values of the two floating-point operands with respect to the ordering

$$-\infty < -\max < \dots < -\min < -0.0 == +0.0 < +\min < \dots < +\max < +\infty$$

where **min** and **max** are the smallest and largest positive finite values that can be represented in the given floating-point format. Notable effects of this ordering are:

- Negative and positive zeros are considered equal.
- A negative infinity is considered less than all other values, but equal to another negative infinity.
- A positive infinity is considered greater than all other values, but equal to another positive infinity.

Lifted (§12.4.8) forms of the unlifted predefined floating-point comparison operators defined above are also predefined.



9.3.5 Simple types

C# provides a set of predefined struct types called the simple types. The simple types are identified through keywords, but these keywords are simply aliases for predefined struct types in the **System** namespace, as described in the table below.

Keyword	Aliased type
sbyte	System.SByte
byte	System.Byte
short	System.Int16
ushort	System.UInt16
int	System.Int32
uint	System.UInt32
long	System.Int64
ulong	System.UInt64
char	System.Char
float	System.Single
double	System.Double
bool	System.Boolean
decimal	System.Decimal

ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Спасибо за внимание! До новых встреч!



Александр Шевчук



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

MCID: 9230440

Информационный видеоресурс для разработчиков программного обеспечения

