

Појам исказа

Исказ (statement) је команда којом се обавља нека акција као што је рачунање вредности, чување резултата или приказивање поруке кориснику.

Коришћењем више исказа се праве методе, тј може се рећи да су методе именовани низ исказа (метода Main).

Искази у C# следе јасно дефинисани скуп правила који описују њихов облик и конструкцију.

Ова правила су позната под називом синтакса.

Опис онога што искази раде је познато под називом семантика.

Једно од основних синтаксних правила у C# језику јесте да се сви искази морају завршити са ;.

Програмски језик C# је језик слободног формата, што значи да празнине, знак за размак или нови ред нису битни јер не утичу на реализацију кода и имају само естетски значај.

Употреба идентификатора

Идентификатори (identifiers) су називи који се користе за идентификовање елемената у програмима као што су простор имена, класе, методе и променљиве.

Правила при бирању назива идентификатора:

- Могу се користити само слова (велика и мала), цифре и доње црте (underscore)
- Идентификатор мора да почне словом, доњом цртом или са @
- Не смеју се користити већ резервисани идентификатори

Кључне речи

У C# је резервисано 77 идентификатора за посебне намене и идентификаторе се не могу користити као сопствени.

Ови идентификатори се зову кључне речи (keywords, службене речи) и сваки има посебно значење.

Кључне речи су по дифолту обојене у плаво у Visual Studio коду:

abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	goto
if	implicit	in	int	interface
internal	is	lock	long	namespace
new	null	object	operator	out
override	params	private	protected	public
readonly	ref	return	sbyte	sealed
short	sizeof	stackalloc	static	string
struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked
unsafe	ushort	using	var	virtual
void	volatile	while		

У C# језику се користе идентификатори који нису резервисани што значи да се могу користити као идентификатори за методе, промењиве и класе али је савет да се не користе:

add	alias	async	await	dynamic
get	global	nameof	partial	remove
set	value	when	where	yield
ascending	by	descending	equals	from
group	in	into	join	let
on	orderby	select	where	

Промењиве

Промењива (variable) је место у меморији где се чува нека вредност.

Свакој промењивој у програму се мора дати једнозначни назив којим се на јединствен начин препознаје у окружењу у коме се користи.

Назив промењиве се користи за указивање на вредност коју садржи.

Препоруке за именовање промењивих:

- Не започињите идентификатор промењиве са доњом цртом, иако јесте дозвољено, пошто се онда такав код не може користити код других програмских језика
- Не правити идентификаторе у истом коду који се разликују само по великом или малом слову, пошто постоје програмски језици који то не разликују (Visual Basic)
- Назив започети малим словом
- У идентификатору са више речи, другу и све остале речи започети са великим словом (камиља нотација, camel case)

Тип података који садржи промењива и назив промењиве се користе у исказу за декларацију промењиве: `int starost;`

`starost` је промењива типа `int` а после декларације може се и иницијализовати (доделити јој вредност): `starost = 30;`

Знак једнакости је оператор доделе (assignment) којим се вредност са његове десне стране додељује промењивој са његове леве стране.

Ако се промењивој не додели нека вредност пре њене употребе, јавља се грешка због правила обавезног додељивања.

Основни типови података

тип	класа	знак	величина	опсег
sbyte	System.Sbyte	Yes	1 byte	–128 to 127
short	System.Int16	Yes	2 bytes	–32768 to 32767
int	System.Int32	Yes	4 bytes	–2147483648 to 2147483647
long	System.Int64	Yes	8 bytes	–9223372036854775808 to 9223372036854775807
byte	System.Byte	No	1 byte	0 to 255
ushort	System.UInt16	No	2 bytes	0 to 65535
uint	System.UInt32	No	4 bytes	0 to 4294967295
ulong	System.UInt64	No	8 bytes	0 to 18446744073709551615
float	System.Single	Yes	4 bytes	–1.5x10 ⁻⁴⁵ to 3.4 x 10 ³⁸
double	System.Double	Yes	8 bytes	–5.0x10 ⁻³²⁴ to 1.7x10 ³⁰⁸
decimal	System.Decimal	Yes	12 bytes	1.0x10 ⁻²⁸ to 7.9x10 ²⁸
char	System.Char		2 bytes	Уникод знаци
boolean	System.Boolean		1 byte	true или false

Литералне вредности (литерали)

У изразу: `int x = 30`; нема проблема пошто је 30 целобројна вредност.

У изразу: `long q = 30L`; мора се назначити да се вредност 30 користи као типа `long integer`.

У изразу: `float z = 30F`; мора се назначити да желимо користити вредност 30 као типа `float`.

У изразу: `double w = 0.3`; није потребно посебно назначити да се вредност 0.30 жели користити са дуплом прецизности.

У изразу: `float z = 0.3F`; мора се назначити да желимо користити вредност 0.3 као типа `float`.

У изразу: `decimal c = 0.3M`; мора се назначити да је 0.3 типа `decimal`.

тип	категорија	суфикс	пример
<code>bool</code>	Boolean		<code>true</code> или <code>false</code>
<code>int</code> , <code>uint</code> , <code>long</code> , <code>ulong</code>	Integer		100
<code>uint</code> , <code>ulong</code>	Integer	у или U	100U
<code>long</code> , <code>ulong</code>	Integer	l или L	100L
<code>Ulong</code>	Integer	ul, uL, Ul, UL, lu, lU, Lu, LU	100UL
<code>float</code>	Real	f или F	1.5F
<code>double</code>	Real	без ознаке, d или D	1.5
<code>decimal</code>	Real	m или M	1.5M
<code>char</code>	Character		'a' или ескејп секвенца
<code>string</code>	String		"a...a", може укључити ескејп секвенце