



CENTAR ZA  
mlade talente

# Primena objektno orijentisanog programiranja u izradi složenih aplikacija (programski jezik C#)

## Uvod u C#

# Uvod



C# je programski jezik razvijen u okviru .NET Framework-a  
Objektno orijentisan (zasnovan na klasama), strukturiran, funkcionalan, sa podrškom za događaje  
Dizajniran i kreiran od strane Microsoft-a

## Kratka istorija

- Objavljen u julu 2000.

- Prva verzija 1.0 se pojavila u 2001.

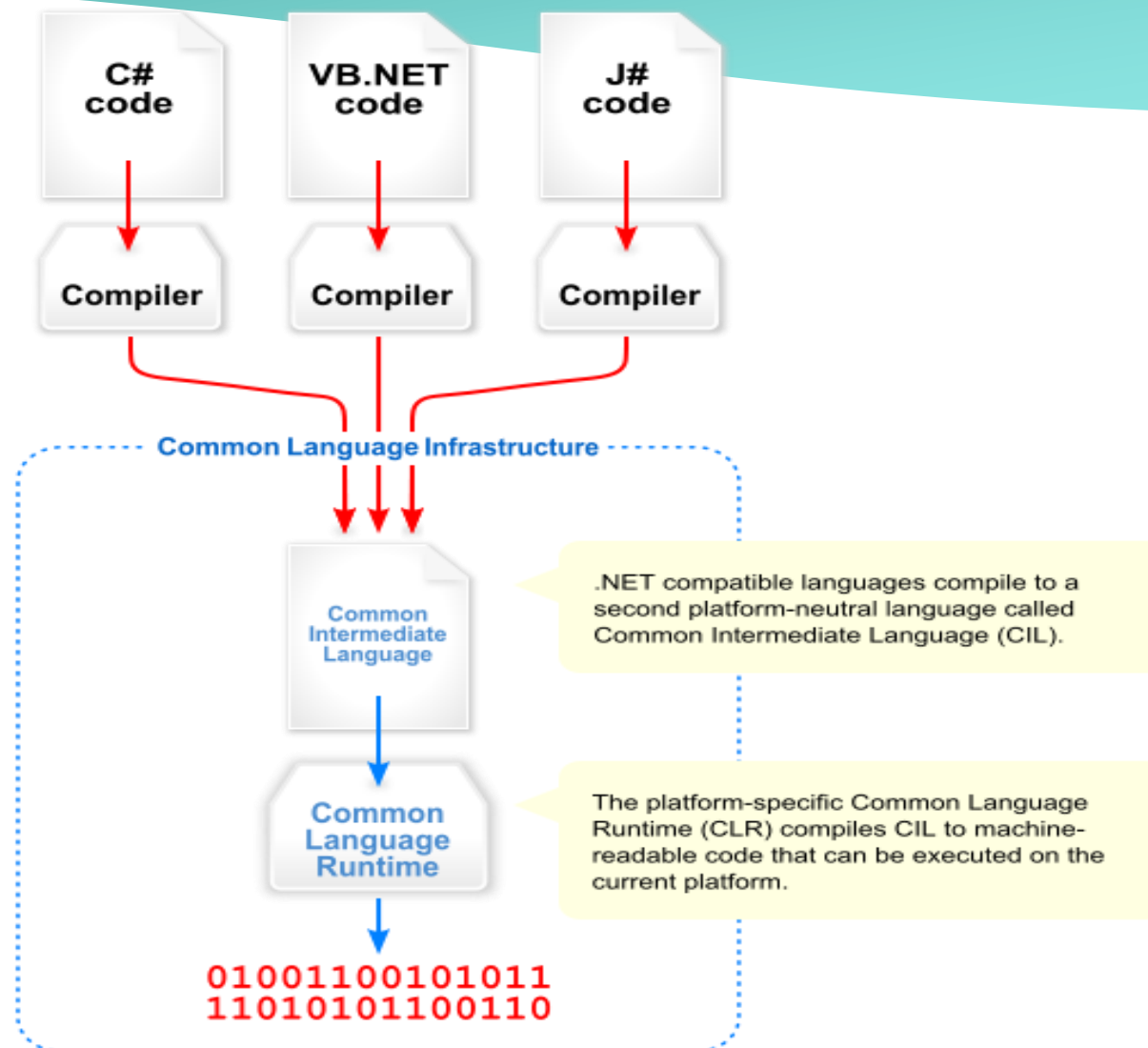
- Standardizovan od strane ISO-a u aprilu 2003.

- Trenutna verzija 9.0

# Ciljevi C#

- Namenjen da bude jednostavan i objektno-orijentisan jezik za razvoj svih vrsta aplikacija
- Web aplikacije, aplikacije sa korisničkim interfejsom, rad sa bazama podataka, itd.
- Vršiti razne bezbednosne provjere kao što su provjera tipova i opsega niza, detektuje pristupanje neinicijalizovanim promenljivima, itd.
- Podrška za internacionalizaciju (globalizacija i lokalizacija)
- Lak prelazak sa C/C++, Java ili objektnog paskala
- Poboľšan i efikasniji razvoj aplikacija
- Ugrađen mehanizam za automatsko upravljanje memorijom

# Grafički prikaz kompajliranja koda



# Imenski prostor

- Izvršni kod u .NET-u je organizovan i grupisan po imenskim prostorima (eng. *namespace*)
- Omogućava izbegavanje konflikata u imenima elemenata dva različita skupa izvornih kodova
- Imenski prostori se ne odnose na imena fajlova ili direktorijuma
- Izraz '**using**', zajedno sa imenom imenskog prostora, omogućava pristup i korišćenje članove i funkcionalnosti iz tog prostora
- Sa druge strane, punim imenom nekog člana imenskog prostora se pristupa tom članu, bez korišćenje '**using**' izraza  
`using System.Text;`

# Struktura izvornog koda

- C# sintaksa je zasnovana na sintaksi C jezika
- Završetak C# izraza se označava sa tačkom-zarez (;)
- Proizvoljan broj izraza se grupišu u blokove
- Izrazi se smeštaju u vitičaste zagrade ({ })
- Metoda **static Main** je ulazna tačka za sve aplikacije i prva je metoda koja se poziva kada se aplikacija pokrene i izvrši
  - Vraća **void** ili **int**
  - Može kao argument da prima **niz stringova** ili da nema **nijedan** argument
  - Fajlovi izvornog koda imaju ekstenziju **.cs**
  - MSVS projektni fajlovi imaju ekstenziju **.csproj**

# Tipovi podataka

Tipovi podataka se grupišu u dve kategorije:

- 'Value' tipovi
  - Osnovni tipovi (int, float, bool...)
  - Enumeracije
  - Struktture
  - Nullable vrednosti
- 'Reference' (Referentni) tipovi
  - Klase
  - Interfejsi
  - Delegati
  - Nizovi

# Tipovi podataka

Takođe se mogu grupisati i na sledeći način :

## **Predefinisane** tipove

- ugrađeni tipovi koji su već definisani i poznati kompajleru.
- Postoje primitivni ugrađeni tipovi kao što su *integer* ili *boolean*, i kompleksni ugrađeni tipovi kao što su string-ovi

## **Korisnički definisani** tipovi

- deklarirani od strane korisnika



# Predefinisani tipovi – numerički tipovi podataka

Tip	Opis	Opseg [od]	Opseg [do]
<b>byte</b>	8-bitni neoznačen ceo broj	0	255
<b>sbyte</b>	8-bitni označen ceo broj	-128	127
<b>short</b>	16-bitni označen ceo broj	-32.768	32.767
<b>ushort</b>	16-bitni neoznačen ceo broj	0	65.535
<b>int</b>	32-bitni označen ceo broj	-2.147.483.648	-2.147.483.647
<b>uint</b>	32-bitni neoznačen ceo broj	0	-4.294.967.295
<b>long</b>	64-bitni označen ceo broj	-9.223.372.036.854.775.808	9.223.372.036.854.775.807
<b>ulong</b>	64-bitni neoznačen ceo broj	0	18.446.744.073.709.551.615
<b>float</b>	32-bitni realni broj	-3,402823 e38	3,402823 e38
<b>double</b>	64-bitni realni broj	-1,79769313486232 e308	1,79769313486232 e308
<b>decimal</b>	128-bitni realni broj	-1,79769313486232 e308	1,79769313486232 e308

# Predefinisani tipovi – alfanumerički tipovi podataka

## char

- 16-bitni Unicode karakter
- Može da sadrži samo vrednost jednog karaktera (npr 'a' 'B' '(' '0')
- Može da sadrži sve znakove (slova, brojeve, specijalne karaktere)

## string

- Sekvenca Unicode karaktera
- 'teams sastanak' , 'polaznik'...

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	!	64	40	@	96	60	'
^A	1	01		SOH	33	21	!"	65	41	A	97	61	a
^B	2	02		STX	34	22	!#"	66	42	B	98	62	b
^C	3	03		ETX	35	23	!#" \$	67	43	C	99	63	c
^D	4	04		EOT	36	24	!#" \$ %	68	44	D	100	64	d
^E	5	05		ENQ	37	25	!#" \$ % &	69	45	E	101	65	e
^F	6	06		ACK	38	26	!#" \$ % & '	70	46	F	102	66	f
^G	7	07		BEL	39	27	!#" \$ % & ' (	71	47	G	103	67	g
^H	8	08		BS	40	28	!#" \$ % & ' ( )	72	48	H	104	68	h
^I	9	09		HT	41	29	!#" \$ % & ' ( ) *	73	49	I	105	69	i
^J	10	0A		LF	42	2A	!#" \$ % & ' ( ) * +	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	!#" \$ % & ' ( ) * + ,	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	!#" \$ % & ' ( ) * + , -	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	!#" \$ % & ' ( ) * + , - .	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	!#" \$ % & ' ( ) * + , - . /	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	!#" \$ % & ' ( ) * + , - . / 0	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	!#" \$ % & ' ( ) * + , - . / 0 1	80	50	P	112	70	p
^Q	17	11		DC1	49	31	!#" \$ % & ' ( ) * + , - . / 0 1 2	81	51	Q	113	71	q
^R	18	12		DC2	50	32	!#" \$ % & ' ( ) * + , - . / 0 1 2 3	82	52	R	114	72	r
^S	19	13		DC3	51	33	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4	83	53	S	115	73	s
^T	20	14		DC4	52	34	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5	84	54	T	116	74	t
^U	21	15		NAK	53	35	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6	85	55	U	117	75	u
^V	22	16		SYN	54	36	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7	86	56	V	118	76	v
^W	23	17		ETB	55	37	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8	87	57	W	119	77	w
^X	24	18		CAN	56	38	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9	88	58	X	120	78	x
^Y	25	19		EM	57	39	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 :	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ;	90	5A	Z	122	7A	z
^[	27	1B		ESC	59	3B	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; <	91	5B	[	123	7B	{
^\	28	1C		FS	60	3C	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < =	92	5C	\	124	7C	
^]	29	1D		GS	61	3D	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = >	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?	94	5E	^	126	7E	~
^-	31	1F	▼	US	63	3F	!#" \$ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?	95	5F	_	127	7F	␣

# Predefinisani tipovi podataka

## **Bool**

- 8-bitna logička vrednost
- Logički tip podataka čija vrednosti može biti true (tačna) ili false (netačna)

## **Object**

- Bazni tip svih ostalih promenljivih

## **DateTime**

- Tip podataka koji prikazuje vreme

# 'Value' tipovi

- Direktno sadrže vrednosti podataka gde su alocirani
- Mogu biti ugrađeni tipovi ili korisnički definisani tipovi
- Ne mogu da naslede druge tipove niti se mogu naslediti
- Ne mogu imati podrazumevani konstruktor
- Instance se mogu kreirati tako što se samo deklarišu
- Inicijalizuju se primenom operatora **new** ili im se **direktno** dodeli vrednost tokom ili nakon kreiranja
- Ako se samo deklarišu, moraju se ručno inicijalizovati pre nego što im se pristupi

# 'Reference' tipovi

- Sadrže memorijske adrese (reference) podataka
- Mogu biti korisnički definisani tipovi
- Podrazumevano se upoređuju po referencama a ne po vrednostima podataka na koje reference pokazuju
- Podrazumevano se vrši kopiranje po referencama a ne po vrednostima
- Instance se kreiraju i inicijalizuju primenom operatora **new**
- Njihova podrazumevana vrednost je nula (nedefinisana) vrednost

# Promenljive



- Promenljive su identifikatori koji predstavljaju memorijske lokacije gde se čuvaju vrednosti
- Mogu da postoje isključivo kao članovi klasa/struktura ili kao lokalne promenljive u metodama, ali ne kao globalne promenljive
- Svaka promenljiva mora da se deklarise pre upotrebe i mora biti inicijalizovana pre nego što joj se pristupi
- Deklarišu se tako što se navede tip podatka pa zatim ime promenljive na jedan od sledećih načina:

Samo deklaracija, npr. `int mojInt;`

Deklaracija i inicijalizacija, npr. `int mojInt = 1;`

Deklaracija pa zatim inicijalizacija, npr. `int mojInt; mojInt = 1;`

# Console klasa



- Klasa Console predstavlja standardni ulaz, izlaz i stream za greske konzolnih aplikacija.
- Neke od metoda koje mogu da se pozivaju
  - ReadKey() – metoda koja čita jedan karakter sa konzole
  - ReadLine() – metoda koja čita celu liniju sa konzole
  - Write() – metoda koja ispisuje niz karaktera na konzolu
  - WriteLine() – metoda koja ispisuje niz karaktera na konzolu i na kraju dodaje terminirajući znak za liniju (prelazak na drugi red)
  - Clear() – metoda koji briše sve sa ekrana konzole što je do tada ispisano

**Hvala na pažnji!**

CENTAR ZA  
mlade talente