

Увод у С# - 1

Пројектовање информационих система и база података

Ненад Пантелић

nenadpantelickg@gmail.com, 564.2015@fink.rs

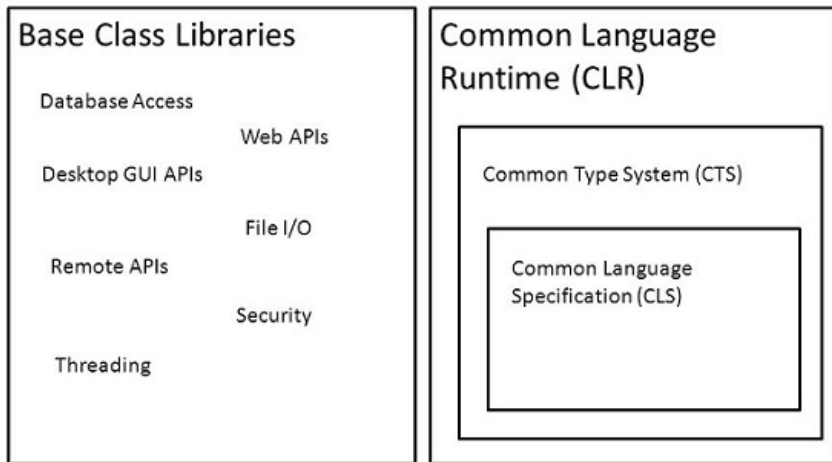
22.октобар 2018.

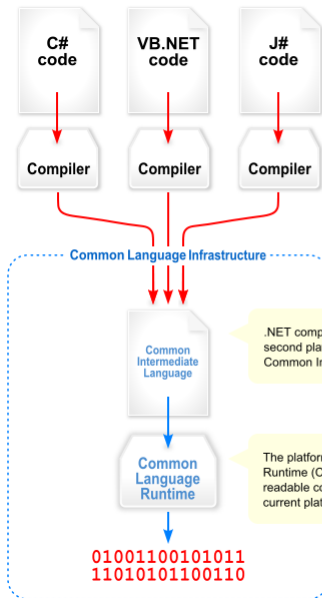
Садржај

- 1 Увод
- 2 Структура програма
- 3 Типови података
- 4 Оператори
- 5 Console I/O

- 1999. Андерс Хејлсберг (енг. *Anders Hejlsberg*) води тим који ради на новом језику - **C#**
- 2000. излази прва верзија језика
- тренутна стабилна верзија је **7.3** (мај 2018.)
- стандардно се користи уз **Microsoft Visual** или **Mono** компајлер
- C# је дизајниран за *Common Language Infrastructure (CLI)* = извршни код + runtime окружење
- CLI је имплементиран код .NET framework - а кроз CLR

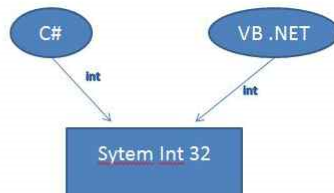
.NET framework





.NET compatible languages compile to a second platform-neutral language called Common Intermediate Language (CIL).

The platform-specific Common Language Runtime (CLR) compiles CIL to machine-readable code that can be executed on the current platform.



CTS

- модеран језик опште намене
- објектно - оријентисан и компонентно - оријентисан језик
- лак за учење и добро структуриран
- ефикасан као део .NET платформе
- може се користити на различитим платформама

C# = Java или C++

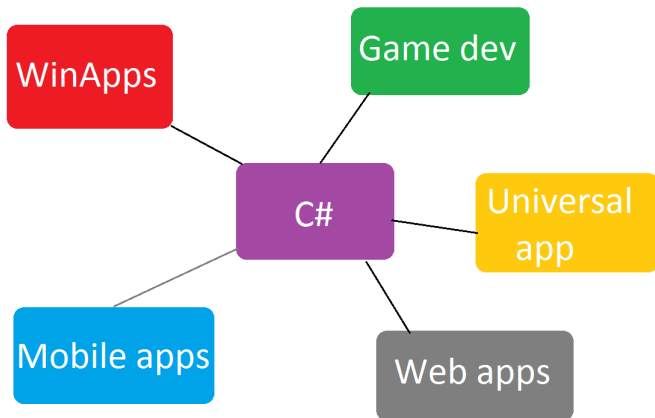
Џејмс Гослинг о C#

"imitation" of Java; sort of Java with reliability, productivity and security deleted.

Клаус Крефт и Анђелика Лангер

Java and C# are almost identical programming languages. Boring repetition that lacks innovation; Hardly anybody will claim that Java or C# are revolutionary programming languages that changed the way we write programs.

Где се C# користи?

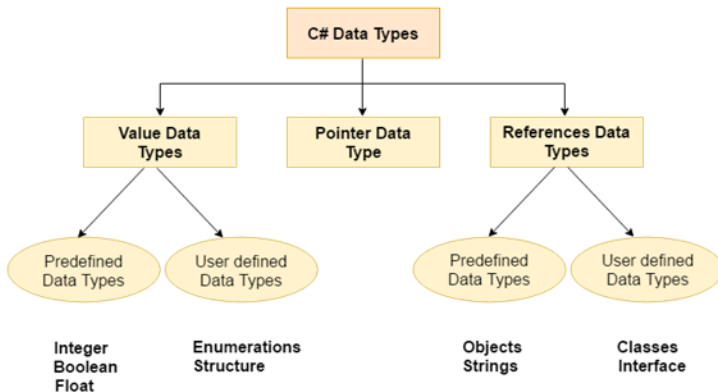


Hello World програм

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApp1
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Hello,world!");
14             Console.ReadLine();
15         }
16     }
17 }
```

- линијски коментари - //
- блок коментари - /**/
- коментари документације
- task list коментари

Типови података



Примитивни типови података

Type	Represents	Range	Default Value
bool	Boolean value	True or False	False
byte	8-bit unsigned integer	0 to 255	0
char	16-bit Unicode character	U +0000 to U +ffff	'\0'
decimal	128-bit precise decimal values with 28-29 significant digits	$(-7.9 \times 10^{28} \text{ to } 7.9 \times 10^{28}) / 10^0$ to 28	0.0M
double	64-bit double-precision floating point type	$(+/-)5.0 \times 10^{-324} \text{ to } (+/-)1.7 \times 10^{308}$	0.0D
float	32-bit single-precision floating point type	$-3.4 \times 10^{38} \text{ to } +3.4 \times 10^{38}$	0.0F
int	32-bit signed integer type	-2,147,483,648 to 2,147,483,647	0
long	64-bit signed integer type	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	0L
sbyte	8-bit signed integer type	-128 to 127	0
short	16-bit signed integer type	-32,768 to 32,767	0
uint	32-bit unsigned integer type	0 to 4,294,967,295	0
ulong	64-bit unsigned integer type	0 to 18,446,744,073,709,551,615	0
ushort	16-bit unsigned integer type	0 to 65,535	0

Runtime типови

Reserved word	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

- аритметички оператори
- релациони оператори
- логички оператори
- битовски оператори
- оператори доделе
- остали оператори

Аритметички и релациони оператори

```
int a = 10, b = 20;

//aritmeticki operatori
Console.WriteLine(a + b); // 30
Console.WriteLine(a - b); // -10
Console.WriteLine(a * b); // 200
Console.WriteLine(a / b); // 0
Console.WriteLine(a % b); // 10
Console.WriteLine(a++); // 10
Console.WriteLine(++a); //12
Console.WriteLine(a--); //12
Console.WriteLine(--a); //10

a = 10;
b = 20;

//relacioni operatori
Console.WriteLine(a > b); // False
Console.WriteLine(a < b); // True
Console.WriteLine(a >= b); // False
Console.WriteLine(a <= b); // True
Console.WriteLine(a == b); // False
Console.WriteLine(a != b); // True
```

Логички и битовски оператори

```
bool c = true;
bool d = false;

Console.WriteLine(c && d); // False
Console.WriteLine(c || d); // True
Console.WriteLine(!c); // False
Console.WriteLine(c ^ d); // True

a = 60; // 0011 1100
b = 13; // 0000 1101

Console.WriteLine(a & b); //12 = 0000 1100
Console.WriteLine(a | b); //61 = 0011 1101
Console.WriteLine(~a); //-61 = 1100 0011
Console.WriteLine(a ^ b); //49 = 0011 0001
Console.WriteLine(a << 2); //240 = 1111 0000
Console.WriteLine(a >> 2); //15 = 0000 1111
```


Оператори доделе и типовски оператори

```
//operatori dodele
a = 20;
a += 10; //30
a -= 10; //20
a *= 10; //200
a /= 10; // 20
a %= 6; //2
a <<= 2; // 8
a >>= 2; // 2
a &= 4; //0
a |= 4; //4
a ^= 4; // 0

//operatori koji se ticu tipa podataka
Console.WriteLine(typeof(int)); // System.Int32
Console.WriteLine(a.GetType()); // System.Int32
Console.WriteLine(sizeof(int)); // 4
```

Nullable типови - 1

```
// kompajler prijavljuje gresku
int a = null;

//ovo je nullable tip
int? b = null;
Console.WriteLine(b); //stampa praznu liniju

if (b != null)
    Console.WriteLine("Ima vrednost");
else
    Console.WriteLine("Null vrednost");
```

Nullable типови - 2

```
b = null; //3
/* False; ako je vrednost b = 3, onda je True*/
Console.WriteLine(b.HasValue);
/*
 * ovo dize izuzetak - nije moguće pristupiti property-ju Value
 * nullable tipa ako je vrednost promenljive null.
 * Ako je vrednost prethodnog izraza 3, nema izuzetka.
 */
Console.WriteLine(b.Value);

/* kako uraditi */
// ako postoji vrednost, pristupamo joj
if (b.HasValue)
    Console.WriteLine(b.Value);
/*
 * ako je vrednost nullable tipa razlicita od null,
 * koristi se ta vrednost, inace se koristi podrazumevana
 * vrednost tog tipa za datu promenljivu
 */
Console.WriteLine(b.GetValueOrDefault());
/* ako je vrednost promenljive razlicita od null,
 * koristi se ta vrednost, inace se koristi vrednost
 * koja je navedena iza operatora ??
 */
Console.WriteLine(b ?? 5);
```

Console I/O

```
Console.WriteLine("Primer stampe");

int a = 2, b = 5, c = 7;
// brojevi u viticastim zagrada predstavljaju indekse argumenata,
// ukoliko je unet indeks nepostojećeg argumenta, podize se exception
Console.WriteLine("num1 = {0}, num2 = {1}, num3 = {2}, num4 = {1}, num5 = {0}", a, b, c);
//dize se format exception ovde - ne postoji argument sa indeksom 3
Console.WriteLine("num1 = {0}, num2 = {1}, num3 = {2}, num4 = {1}, num5 = {3}", a, b, c);

//vraca procitanu liniju kao string
string str = Console.ReadLine();
Console.WriteLine(str);

// cita karakter sa konzole, argument true u metodi ReadKey,
// ne prikazuje uneti karakter, samo ga cita - bez echo, vraca kao karakter
char d = Console.ReadKey().KeyChar;
//stampa bez prelaza u novi red
Console.Write(d);
Console.Write("Jos jedan ispis");

/* brise sadrzaj iz konzole*/
Console.Clear();
//cita jedan karakter sa unosa - vraca int
int x = Console.Read();
Console.WriteLine(x);
Console.ReadLine();
```