



**SVEUČILIŠTE U ZAGREBU**  
**Fakultet prometnih znanosti**  
**Zavod za inteligentne transportne sustave**  
**Vukelićeva 4, Zagreb, HRVATSKA**



# **Računalstvo**

## **Uvod u C# programiranje**

**Doc. dr. sc. Edouard Ivanjko, dipl.ing.**

# Sadržaj

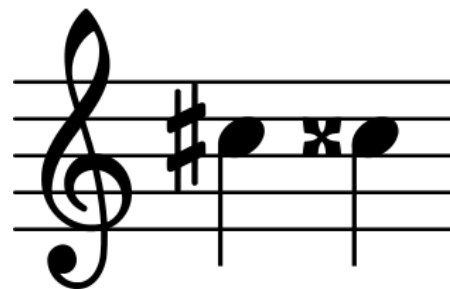
---

- Programski jezik C#
- Opća pravila pisanja C# programa
- Ključne riječi
- Struktura C# programa
- Pretprocesorske direktive
- Komentiranje programa
- Primjeri



# Programski jezik C#

- Izgovara se “si šarp”
- Inspiracija imenu je došla od povisilice iz pravila pisanja nota
  - C -> C++ -> C#
- Nastao 2000. godine
  - Proizvođač Microsoft
  - Voditelj projekta Anders Hejlsberg
  - U primjeni od 2001. godine
  - Specifikacija C# 4.0
    - <http://www.ecma-international.org/publications/standards/Ecma-334.htm>



# Programski jezik C#

- Osnovne karakteristike
    - Viši programski jezik, nadogradnja jezika C++
    - Jednostavan, objektno orijentiran
    - Za općenitu primjenu, prenosiv
    - Strukturirani i imperativni programski jezik
  - Razvojne okoline
    - SharpDevelop
    - MonoDevelop
    - Borland Turbo C#
    - MS Visual Studio
- MS Visual Studio Express C# -> besplatna inačica



# Opća pravila pisanja C# programa

- Kôd se unosi u tekstualne datoteke
  - Nastavak datoteke “cs”
- Podržani samo znakovi engleske abecede
- Razlikuju se velika i mala slova
- Mjesto početka naredbe u retku je proizvoljno
- Kraj naredbe se označava znakom “;”
- Dopušteno je više naredbi u istom retku
  - `int a; string boja = “zelena”; a = boja.Length();`
- Poželjno umetanje praznina i praznih redova



# Opća pravila pisanja C# programa

---

- Slobodan format (stil pisanja nije propisan)
- Programski blokovi označavaju se vitičastim zagradama
  - “{” početak programskog bloka
  - “}” kraj programskog bloka
- Programski blok obuhvaća
  - Deklaracije
  - Definicije
  - Druge programske blokove
  - Pozive funkcija/metoda



# Opća pravila pisanja C# programa

- Davanje imena varijablama i metodama
  - Ne smije se koristiti isto ime za više različitih varijabli ili metoda
    - Imenički prostor (ključna riječ “namespace”) definira područje valjanosti varijable odnosno imena varijable
  - Paziti na smislenost imena
    - varijabla1, metoda2 -> loša imena
  - Kod imena varijabli prvo slovo malo, a svaka sljedeća riječ velikim slovom
    - prviPribrojnik, brojiloStudentata -> dobra imena
  - Kod imena klase ili metoda prvo slovo veliko kao i svaka sljedeća riječ
    - SenzorGPS, TeretnaKompozicija -> dobra imena



# Ključne riječi

---

- Predefinirani identifikatori s posebnim značenjem za prevodioca
  - Definicija varijabli
  - Vraćanje rezultata izvođenja funkcije
  - Operacije usporedbe
  - Definicija petlji
- Pišu se malim slovima
- Ključne riječi se ne smiju koristiti kao imena
  - Varijabli
  - Funkcija
  - Metoda





# Ključne riječi – Vrste

---

- Definicija tipa podatka
  - int, float, double, char, string, short, long
- Modifikatori pristupa varijabli ili metodi
  - private, internal, protected, public
- Naredbe petlji
  - while, for, do
- Naredbe grananja/usporedbe
  - if, else, switch, case
- Naredbe upravljanja izvođenja programa
  - continue, break, exit



# Ključne riječi – Popis

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



# Struktura C# programa – Općenito

- Općenita struktura programa
  - Pretprocesorske naredbe
    - Definicija konstanti
    - Uključivanje definicija potrebnih metoda
  - Definicija pomoćnih metoda
  - Definicija glavne metode
    - Deklaracija varijabli
    - Inicijalizacija varijabli
    - Unos/dohvat podataka
    - Obrada podataka
      - Pozivanje pomoćnih funkcija
    - Ispis podataka/rezultata



# Struktura C# programa – Osnovni kostur

```
// A skeleton of a C# program
using System;
namespace YourNamespace
{
    class YourClass
    {
    }

    struct YourStruct
    {
    }

    interface IYourInterface
    {
    }

    delegate int YourDelegate();

    enum YourEnum
    {
    }

    namespace YourNestedNamespace
    {
        struct YourStruct
        {
        }
    }

    class YourMainClass
    {
        static void Main(string[] args)
        {
            //Your program starts here...
        }
    }
}
```

← Pretprocesorske direktive

← Definicija imeničkog prostora

← Definicija klase

← Definicija strukture

← Definicija sučelja

← Definicija delegata

← Definicija izraza s pobrojavanjem

← Dodatni imenički prostor

← Definicija dodatne strukture istog imena u dodatnom imeničkom prostoru

← Definicija glavne klase

← Glavna metoda programa



# Struktura C# programa – Primjer programa

```

using System;
using System.Collections.Generic;
using System.Text;

namespace Zbrajanje
{
    class Program
    {
        static void Main(string[] args)
        {
            // deklaracija varijabli
            int pribrojnik1, pribrojnik2, suma;

            // inijalizacija varijabli
            pribrojnik1 = pribrojnik2 = suma = 0;

            // učitavanje podataka
            Console.WriteLine("Unesite prvi pribrojnik >> ");
            pribrojnik1 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Unesite drugi pribrojnik >> ");
            pribrojnik2 = Convert.ToInt32(Console.ReadLine());

            // obrada podataka
            suma = pribrojnik1 + pribrojnik2;

            // ispis podataka
            Console.WriteLine("Suma pribrojnika iznosi >> ");
            Console.WriteLine(suma);
            Console.ReadLine();
        }
    }
}

```

← Pretprocesorske direktive

← Definicija imeničkog prostora

← Definicija klase

← Glavna metoda programa


← Deklaracija potrebnih varijabli

← Inicijalizacija početnih vrijednosti varijabli

← Unos podataka

← Obrada podataka

← Ispis rezultata obrade

 **Zbrajanje**



# Pretprocesorske direktive

- Omogućuju upravljanje razvojnom okolinom
  - Kreiranje više inačica programa uvjetnim prevođenjem izvornog kôda
- Vrste
  - Definijske
    - Deklariranje identifikatora      `#define IME`
    - Uklanjanje identifikatora      `#undef IME`
  - Uvjetne -> kreiranje grananja
    - `#if UVJET`      -> ako uvjet ispunjen
    - `#elif UVJET`      -> inače ako uvjet ispunjen
    - `#else`      -> inače
    - `#endif`



# Pretprocesorske direktive

---

- Vrste
  - Pogreške
    - #error PORUKA
  - Brojevi linija
    - #line BROJ\_LINIJE
- Direktiva “**using**”
  - Pozivanje metode bez navođenja punog imena
    - using System; -> skup osnovnih metoda
    - using System.Collections; -> kolekcija klasa podatkovnih struktura -> liste, polje bitova, stog, ...
    - using System.IO; -> skup metoda za unos/ispis u datoteke
    - using System.Text; -> skup metoda za niz znakova



# Pretprocesorske direktive – using System

- Korištenje osnovno potrebnih metoda
- Obuhvaća
  - Rad s konzolom (engl. “command prompt”)
    - Unos podataka
    - Pretvorbu (konverziju) podataka
      - Svi podaci kod unosa su znakovi
      - Pretvorba u potreban format: broj (cijeli ili decimalni), logički iznos, niz znakova
    - Ispis podataka
    - Podešavanje konzole
  - Poveznice s drugim imeničkim prostorima
  - Obrada iznimki





# Komentiranje programa

---

- Olakšavaju čitanje programskog kôda
  - Dokumentacija
    - Podržan XML format
  - Objašnjenje rada programa
- Prevodilac taj dio preskače
- Jednolinijski komentari
  - Protežu se samo unutar jedne linije
  - Mjesto početka komentara nebitno
  - Početak se označava “//”
  - Završetak komentara je znak za novi redak
  - Primjer: // Ovo je jednolinijski komentar



# Komentiranje programa

---

- Višelinijski komentari
  - Sadrži jednu ili više opisnih linija
  - Gniježđenje višelinijških komentara nije dozvoljeno
  - Početak se označava `“/*”`
  - Kraj se označava `“*/”`
  - Sve između prevodilac preskače
  - Primjer

`/*`

Ovo je primjer  
višelinijškog komentara

`*/`

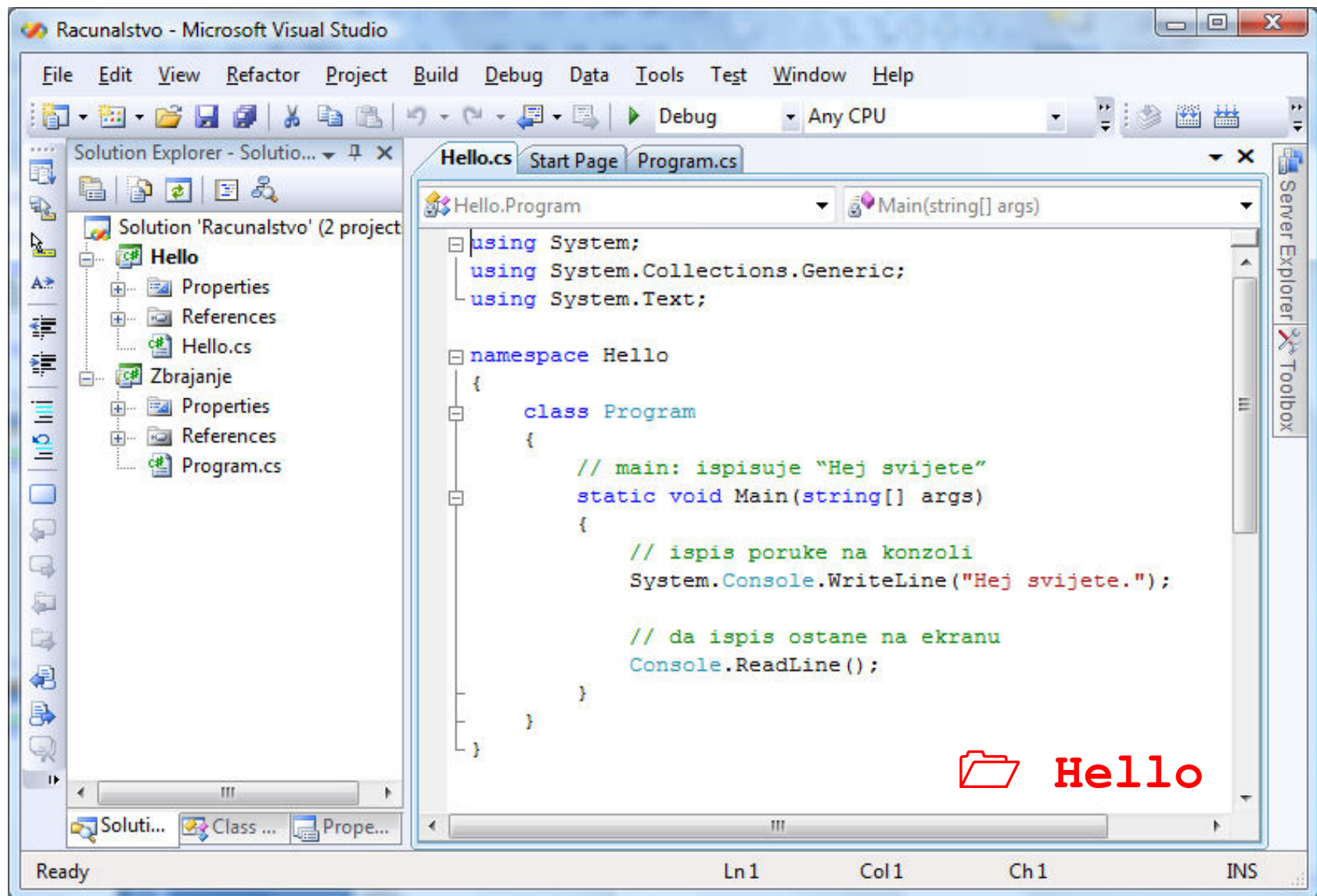


# Komentiranje programa

- XML dokumentacijski komentari
  - Skup jednolinijskih komentara omeđenih tagovima
  - Početak se označava znakom “*///*”
  - Podržani su XML dokumentacijski tagovi
  - Primjer sažetka
 

<i>///</i> <summary>	<code>&lt;c&gt;</code>	<code>&lt;code&gt;</code>	<code>&lt;example&gt;</code>	<code>&lt;exception&gt;</code>
	<code>&lt;list&gt;</code>	<code>&lt;param&gt;</code>	<code>&lt;paramref&gt;</code>	<code>&lt;permission&gt;</code>
	<code>&lt;remarks&gt;</code>	<code>&lt;returns&gt;</code>	<code>&lt;see&gt;</code>	<code>&lt;value&gt;</code>
	<code>&lt;seealso&gt;</code>	<code>&lt;summary&gt;</code>		
  - Mogućnost automatskog stvaranja XML dokumentacije
    - Naredba: “*csc /doc:IME.xml IME.cs*”

# Primjeri – Prvi program “Hej svijete!”



# Primjeri – Jednostavan Internet preglednik

The screenshot displays the Visual Studio IDE with a Windows form named 'Form1.cs [Design]'. The form contains a 'WebBrowser' control and a 'Kreni' button. The 'WebBrowser Tasks' context menu is open, showing the option 'Dock in parent container'. The Properties window on the right shows the 'webPreglednik System.' control with a size of 607x385 pixels.

Property	Value
ScrollBarsEr	True
Size	607; 385
Width	607
Height	385
TabIndex	0
TabStop	True
Tag	
Url	
Visible	True
WebBrowse	True

**Internet Preglednik**