

Osnove programiranja

Promenljive

The slide features a light blue gradient background at the top. Below the main title, there are several thick, wavy, light gray lines that sweep across the lower right portion of the slide, creating a sense of movement and depth.

Sadržaj

- Elementi c# jezika
- Deklarisanje promenljivih
- Konvencije imenovanja promenljivih
- Dodeljivanje vrednosti promenljivama
- Prosti tipovi promenljivih
 - Celobrojne promenljive
 - Promenljive namenjene decimalnim brojevima
 - Char, bool i string
- Operatori i izrazi

Elementi jezika

- To su: azbuka, identifikatori, rezervisane reči, literali, graničnici, operatori, komentari, konstante i promenljive.
- *Azbuka* se sastoji od:
 - Velikih i malih slova engleske abecede;
 - cifara dekadnog brojnog sistema;
 - specijalnih znakova: () + - * ! # % ^ & _ | = ' " : ;
> < , . / \ ?

Elementi jezika

- *Identifikatori* su imena koja se dodeljuju promenljivama, klasama, metodama itd.
 - Case sensitive
- *Rezervisane reči* - imena koje je rezervisao kompajler i ne mogu se koristiti kao identifikatori u programima.

Elementi jezika

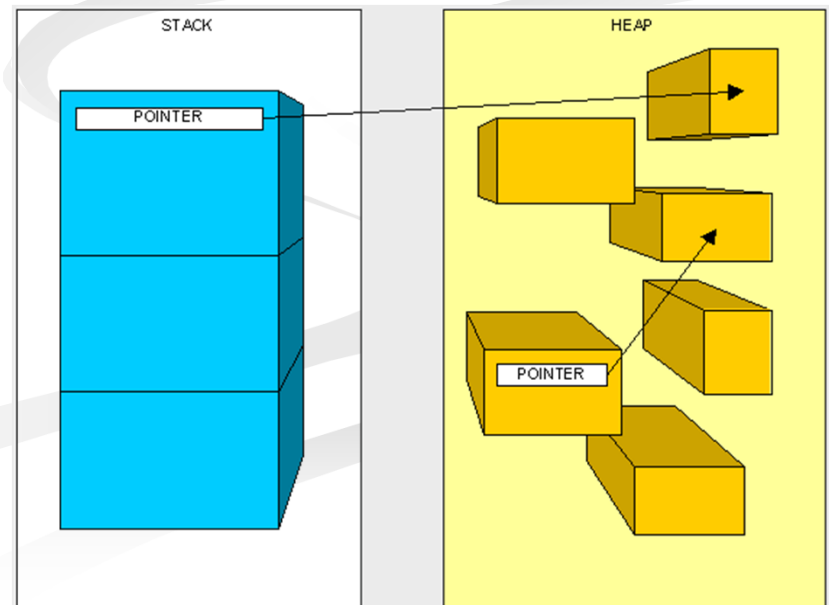
- *Literali* su jedinice podataka umetnute u program.
 - Numerički (celobrojni ili decimalni), znakovni (jedan ili više karaktera), logički (true, false),...
- *Graničnici* služe za razgraničavanje pojedinih delova strukture programa – {...}
- *Operator* transformiše i kombinuje izraze.
- *Konstantne promenljive* – *konstante* su vrednosti koje se zadaju jedanput i ne menjaju se tokom izvođenja programa.

Promenljive

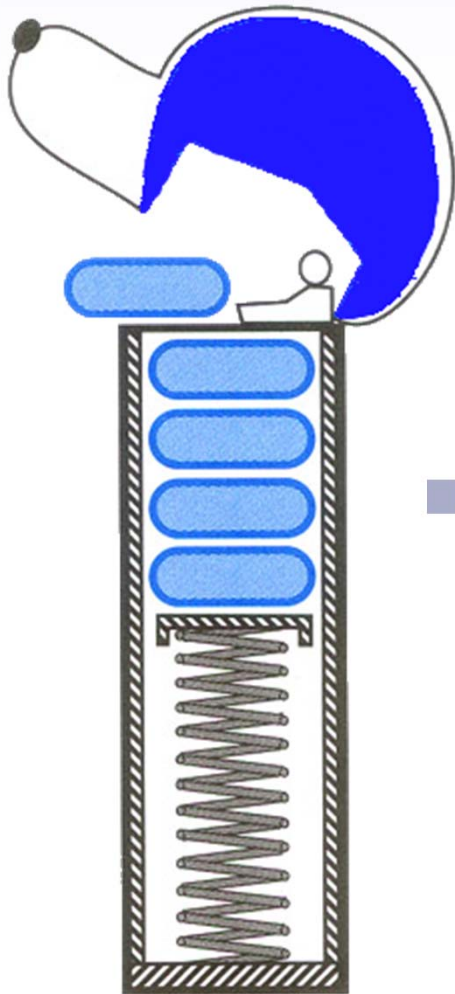
- **Promenljiva** je **ime** (identifikator) memorijske lokacije u kojoj aplikacija **čuva vrednost**.
- Služe za skladištenje podataka.
- Podela:
 - **Vrednosne** - direktno sadrže podatke;
 - **Referentne** – sadrže reference na podatke - adresu lokacije u dinamičkoj memoriji gde se čuvaju odgovarajući podaci.
 - Osnovna razlika je u načinu njihovog smeštanja u računarsku memoriju.

Stek i hip

- Operativni sistem i jezičke infrastrukture (CLR) dele memoriju za čuvanje podataka:
- **Stek** (engl. stack) i **hip** (engl. heap) i potpuno su različito organizovane.
- Stek i hip su mesta gde se smeštaju promenljive i konstante.



Stek

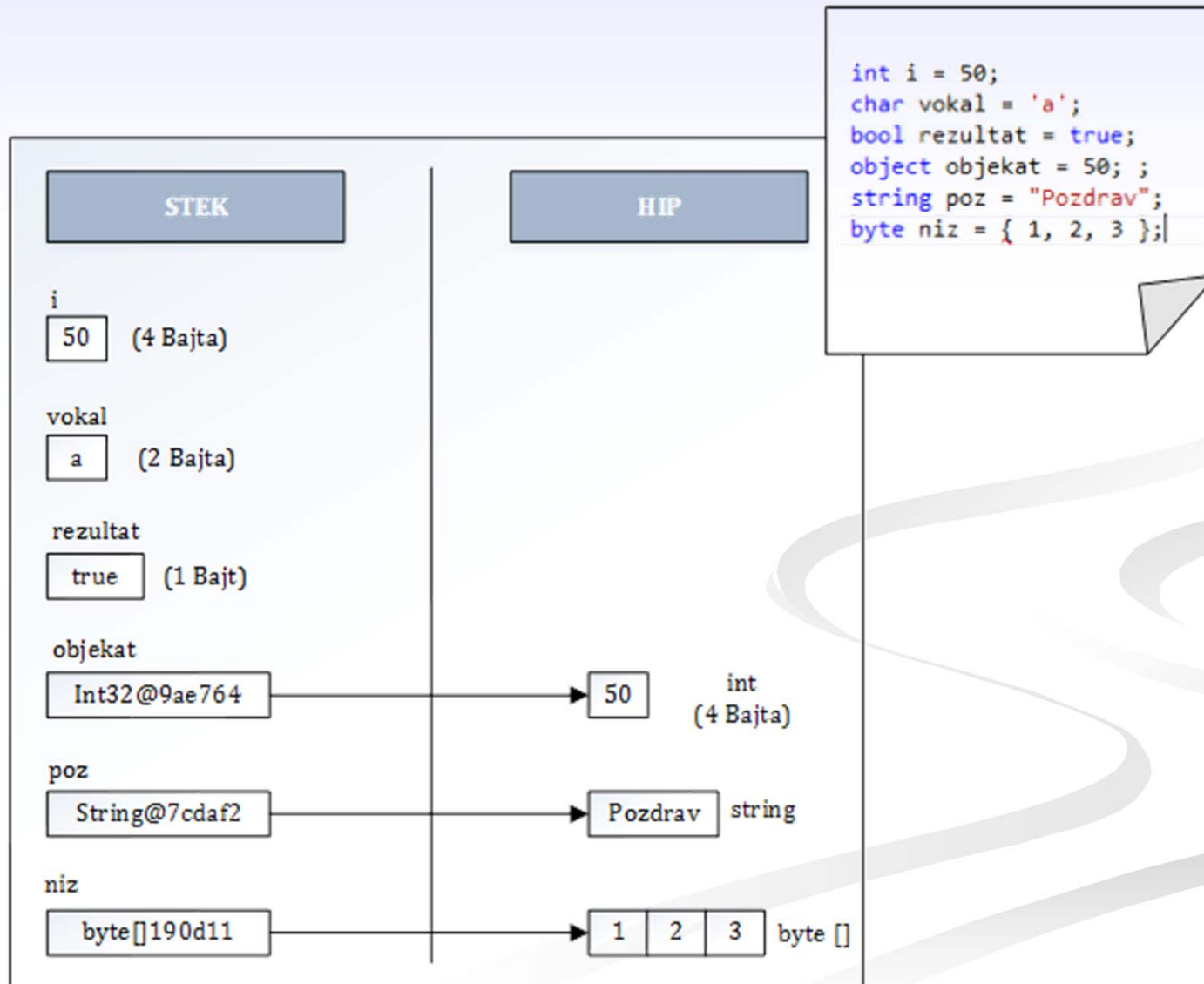


- *Stek* je memorijski blok (memorijska struktura) u koji se smeštaju lokalne promenljive i parametri funkcija (metoda), a organizovan je kao grupa diskova naređanih jedan na drugi.
- On logički raste i sažima se kako funkcija započne, odnosno završi rad (LIFO princip).

Hip

- *Hip* je memorijski blok u koji se smeštaju objekti (tj. instance referentnog tipa).
- Organizovan je kao gomila razbacanih kutija.
 - Svaka kutija ima labelu koja služi za indikaciju njene upotrebljivosti.
- Kad se napravi *nov objekat*, CLR traži *praznu kutiju* na hipu, alocira (*rezerviše*) je za smeštanje tog objekta i *vraća referencu* na taj objekat.
- **Referenca** na objekat se smešta u lokalnu promenljivu na **steku**.

Vrednosne i referentne promenljive



Deklarisanje promenljivih

- Deklarisanje se vrši u obliku:

vidljivost tip naziv = [pocetna vrednost] ;

- Primeri:

- `private int a = 1 ;`

- `public string Ime ;`

- `int x, z ;`

- `decimal plata, porez;`

} vidljivost, OO paradigma

Deklarisanje promenljivih

- Tipom podatka definisan je:
 - skup mogućih vrednosti tog podatka
 - način registrovanja podatka u memoriji
 - skup mogućih akcija nad podatkom.
- C# ima 15 **predefinisanih** tipova:
 - 13 vrednosnih i 2 referentna tipa
- I 2 **korisnički** definisana - nabrajanje, niz.

Tipovi promenljivih

- **Vrednosni tipovi**

- **Numerički**

- Ceo broj sa predznakom (sbyte, short, int, long)
 - Ceo broj bez predznaka (byte, ushort, uint, ulong)
 - Realan broj (float, double, decimal)

- **Logički izraz (bool)**

- **Znak (char)**

- **Nabrojivi tip (enum)**

- **Referentni tipovi**

- **Znakovni niz (string)**

- **Objekat (object)**

- **Niz (array)**

Korisnički
definisana

Imenovanje promenljivih

- Osnovna pravila pri imenovanju promenljivih su:
 - Prvi znak imena promenljive mora biti ili **slovo**, ili znak **_**, ili **@**.
 - Sledeći znak je slovo, znak podvlačenja ili broj.
 - Ključne reči (using, namespace i tome sl.) ne mogu da se koriste

myBigVar
Sve-je-gotovo
@System

Koje IME
je ispravno?

25Maj
using_test
namespace

mojaPromenljiva ≠ MojaPromenljiva

Konvencije imenovanja

- **camelCase** - promenljive
 - Imena promenljivih sastavljena su iz više reči i svaka reč počinje velikim slovom, dok su sva ostala mala, ali je prvo slovo kompletnog imena promenljive malo. Npr. godiste, imeUcenika.
- **PascalCase** - metode
 - Imena promenljivih sastavljena su iz više reči i svaka reč počinje velikim slovom, dok su sva ostala mala. Npr. Godiste, ImeUcenika, DatumUpisa.

Određivanje i dodeljivanje vrednosti promenljivoj

- **Deklarisanje** promenljive vrši se korišćenjem njenog tipa i imena, na primer:
 - `int godište;`
- **Dodeljivanje** vrednosti vrši se uz pomoć operatora dodele =
 - `godište = 25;`
- **Upamtite:** promenljive moraju biti **inicijalizovane** pre nego što se upotrebe – mora im se dodeliti vrednost.

Prosti tipovi promenljivih

- Tu spadaju **numericke, znakovne ili logicke vrednosti**.
- Za celobrojne vrednosti uzima se određen broj bitova čime se predstavlja broj u binarnom formatu.
 - Promenljiva koja smešta N bitova dozvoljava prikaz bilo kog broja između 0 i $2^N - 1$.

Lista celobrojnih tipova

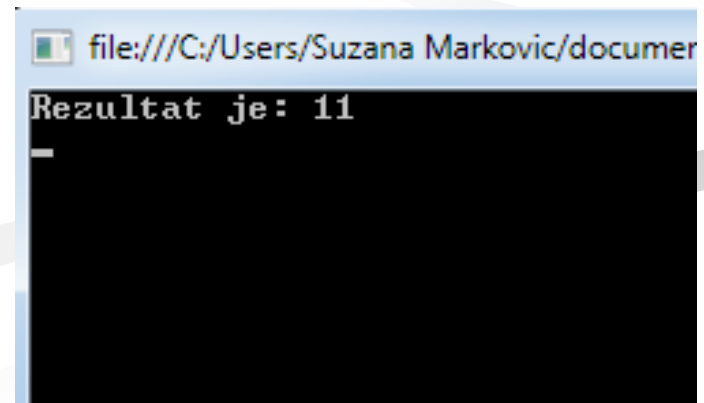
TIP	PSEUDONIM ZA	DOZVOLJENE VREDNOSTI
sbyte	System.sbyte (1)	ceo broj između -128 i 127.
byte	System.byte (1)	ceo broj između 0 i 255.
short	System.int16 (2)	ceo broj između -32768 i 32767.
ushort	System.uint16 (2)	ceo broj između 0 i 65535.
int	System.int32 (4)	ceo broj između -2.147.483.648 i 2.147.483.647.
uint	System.uint32 (4)	ceo broj između 0 i 4.294.967.295.
long	System.int64 (8)	ceo broj između -9223.372.036.854.775.808 i 9223.372.036.854.775.807.
ulong	System.uint64 (8)	ceo broj između 0 i 1846.744.073.709.551.615.

Znak „u“ koji stoji ispred imena nekih promenljivih jeste skraćenica za „unsigned“ (neoznačeno), što znači da se takvim promenljivama ne mogu dodeljivati negativni brojevi.

Primer 1.

- Kreirajte novi c# projekat i nazovite ga **Rad sa promenljivama**.

```
static void Main(string[] args)
{
    int a;
    a = 5;
    int b = 6;
    int rezultat1 = a + b;
    Console.WriteLine("Rezultat je: {0}", rezultat1);
    Console.ReadKey();
}
```



A screenshot of a console window with a blue title bar. The title bar text is "file:///C:/Users/Suzana Markovic/documer". The console output shows "Rezultat je: 11" on the first line, followed by a cursor on the second line.

Šta program radi?

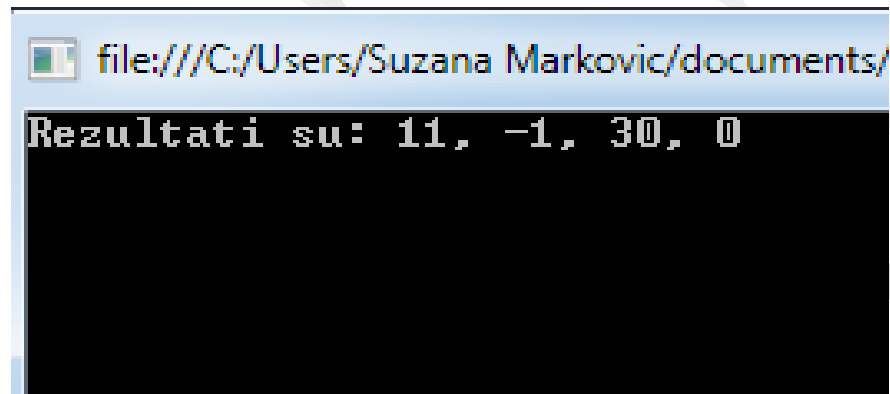
- Deklariše tri promenljive:
 - `int a;`
 - `int b;`
 - `int rezultat1`
- Dodeljuje im vrednosti:
 - `a=5;`
 - `b=6`
 - `rezultat1=a+b`
- Promenljive a i b se sabiraju i njihova vrednost dodeljuje se promenljivoj rezultat1.

Značenje: ("Rezultat je: {0}", rezultat1);

- Svaki znak oivičen vitičastim zagradama predstavlja mesto gde se smešta sadržaj promenljivih.
- Svako mesto (ili string za formatiranje) predstavljeno je celobrojnom vrednošću oivičenom vitičastim zagradama.
- Celobrojne vrednosti počinju od nule i uvećavaju se za jedan, a ukupan broj mesta treba da bude jednak broju promenljivih koje su navedene u produžetku stringa i odvojene su zarezima.
- Kada se tekst ispiše na konzoli, svako mesto biva zamenjeno odgovarajućom vrednošću predviđene promenljive.
- U primeru {0} je zamenjena stvarnom vrednošću promenljive rezultat1.

Primer 2.

```
int a = 5;
int b = 6;
int rezultat1 = a + b;
int rezultat2 = a - b;
int rezultat3 = a * b;
int rezultat4 = a / b;
Console.WriteLine("Rezultati su: {0}, {1}, {2}, {3}", rezultat1, rezultat2, rezultat3, rezultat4);
Console.ReadKey();
```



A screenshot of a Windows console window. The title bar shows the file path: file:///C:/Users/Suzana Markovic/documents/. The console output displays the results of the arithmetic operations: "Rezultati su: 11, -1, 30, 0".

```
file:///C:/Users/Suzana Markovic/documents/
Rezultati su: 11, -1, 30, 0
```

Promenljive za deklaraciju decimalnih brojeva

TIP	Sufiks	PSEUDONIM ZA	Dužina	Pribl. min vrednost	Pribl. max vrednost
float	F	System. Single	4 bajta	+/-1.5x10 ⁻⁴⁵	+/- 3.4x10 ³⁸
double	D (opcion)	System.Double	8 bajtova	+/-5.0x10 ⁻³²⁴	+/- 1.7x10 ³⁰⁸
decimal	M	System.Decimal	16 bajtova	+/-1.0x10 ⁻²⁸	+/- 7.9x10 ²⁸

decimal stanjeNaRacunu = 3433.20; // greška !!!

decimal stanjeNaRacunu = 3433.20**M**

float mojBroj = 1.2**F**;

double mojBroj = 1234.5678**D**;
ISTO

double mojBroj = 1234.5678;

Tip double se odlikuje mnogo većom preciznošću od tipa float!

Ostali tipovi

TIP	PSEUDONIM ZA	DOZVOLJENE VREDNOSTI
bool	System.bool (1)	logička vrednost, true ili false
char	System.char (2)	pojedinačan Unicode znak, smešten kao ceo broj između 0 i 65535
string	System.string	Niz znakova

■ Ne postoji gornja granica za niz znakova koji sačinjavaju string - količina memorije koju on zauzima može biti promenljiva.

■ *Poređenje* vrednosti dve promenljive ili *potvrđivanje* nekog unosa podataka, predstavljaju samo dva od mnogo načina primene logičkih promenljivih.

Primer 4. i 5.

```
char slovo1='a';
```

```
char slovo2 = 'A';
```

```
Console.WriteLine("Za slovo a Unicode vrednost je  
{0}, a za slovo A je: {1}", (int)slovo1,(int)slovo2);
```

```
Console.WriteLine($"Za slovo a Unicode vrednost je  
{(int)slovo1}, a za slovo A je: {(int)slovo2}");
```

```
int a = 10, b=15;
```

```
bool c = a==b;
```

```
Console.WriteLine(c);
```

Primer 6.

```
string ime;  
string prezime;  
Console.WriteLine("Unesite vaše ime:");  
ime = Console.ReadLine();  
Console.WriteLine("Unesite vaše prezime:");  
prezime = Console.ReadLine();  
Console.WriteLine("Dobrodošla {0} {1}", ime, prezime);
```

file:///C:/Users/Suzana Markovic

```
Unesite vase ime:  
Marko  
Unesite vase prezime:  
Nikolic
```

Primer 6.

Modifikacija/interpolacija

```
Console.WriteLine("Unesite vaše ime:");  
string ime = Console.ReadLine();  
Console.WriteLine("Unesite vaše prezime:");  
string prezime = Console.ReadLine();  
Console.WriteLine("Dobrodošla "+ime+" "+prezime);
```

```
Console.WriteLine("Unesite vaše ime:");  
string ime = Console.ReadLine();  
Console.WriteLine("Unesite vaše prezime:");  
string prezime = Console.ReadLine();  
Console.WriteLine($"Dobrodošla {ime} {prezime}");
```

Nabrojivi tip

- Nabrajanje omogućava kreiranje skupa konstanti na koje može da se referencira u programu.
- Nabrajanje je vrednosni tip korisnički definisan.
- Primer:

```
enum Dani
```

```
{  
    Ponedeljak, Utorak, Sreda, Cetvrtak, Petak, Subota,  
    Nedelja  
}
```

- Svakom nabrojivom tipu se pridružuje numerički tip koji predstavlja tip vrednosti nabrojivog tipa.