


Osnove programiranja

Operatori

Izrazi

Sadržaj

- Operatori i izrazi
 - Aritmetički operatori dodele
 - Složeni operatori dodele
 - Binani operatori kod stringova
 - Unarni operatori
 - Relacioni operatori dodele
 - Logički operatori
 - Bitovski operatori
 - Prvenstvo dodeljivanja
- 

Operatori i izrazi

- *Izrazi* su delovi koda koji na osnovu proračuna daju neki rezultat.
 - Svaki izraz koji se završi znakom (;) je naredba.
 - Sastoje se od *operanada* i *operatora*.
 - *Operandi* su *nosioci vrednosti* - podaci koji učestvuju u izrazu
 - *operatori* predstavljaju *operacije* koje se vrše.

Operatori

- Prema broju operanada koje operator povezuje:
 - **unarni** operatori – 1 operand;
 - **binarni** operatori - dva operanda;
 - **ternarni** operatori - tri operanda;
- Najviše binarnih
- Samo jedan ternarni - uslovni operator (logički operator).

Aritmetički operatori dodele

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
+	Binarni	$a = b + c;$	a je dodeljena vrednost sume b i c
-	Binarni	$a = b - c;$	a je dodeljena vrednost razlike b i c
*	Binarni	$a = b * c;$	a je dodeljena vrednost proizvoda b i c
/	Binarni	$a = b / c;$	a je dodeljena vrednost deljenja b i c
%	Binarni	$a = b \% c;$	a je dodeljena vrednost ostatka deljenja b i c
+	Unarni	$a = +b;$	a je dodeljena vrednost b
-	Unarni	$a = -b;$	a je dodeljena vrednost -b

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

```
a=+b;
```

```
a+=b;
```

Izračunaj!

Složeni operatori dodele

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
=	binarni	a = b;	a je dodeljena vrednost b
+=	binarni	a+ = b;	a je dodeljena vrednost a + b
-=	binarni	a- = b;	a je dodeljena vrednost a - b
=	binarni	a = b;	a je dodeljena vrednost a * b
/=	binarni	a/ = b;	a je dodeljena vrednost a / b
%=	binarni	a%= b;	a je dodeljena vrednost ostatka deljenja a i b

```
int x =10;
```

```
x+=1;
```

```
x-=1;
```

```
x*=10;
```

```
x/=10;
```

```
x%= 10
```

Izračunaj!

Binarni operator kod stringova

- string isključivo koristi binarni operator **+**.
 - $a = b + c$;
 - a je dodeljena nadovezana vrednost stringova b i c

```
string string1 = " Dobrodošli ";  
string string2 = " u svet C#! ";  
string string3 = string1 + string2;  
System.Console.WriteLine(string3);
```

```
int a = 3, b = 4;  
int c = a + b;  
Console.WriteLine("Zbir brojeva a i b  
je" + " " + c);
```

```
Za a=3 i b=4 rezultat je 7  
Press any key to continue .
```

Kako ovo dobiti?

Primer 1.

- Napisati konzolnu aplikaciju za unos i spajanje 2 stringa

```
string ime, prezime;  
Console.WriteLine("Unesite vaše ime:");  
ime = Console.ReadLine();  
Console.WriteLine("Unesite vaše prezime");  
prezime = Console.ReadLine();  
Console.WriteLine($"Hvala {ime} {prezime} što ste se  
registrovali!");
```


Unarni operatori

- Operatori **inkrementiranja i dekrementiranja**
 - unarni operatori koji se koriste na dva različita načina: ili odmah ispred (prefiks), ili odmah iza operanda (sufiks).

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
++	unarni	a= ++b	a je dodeljena vrednost. b+1 i b je uvećana za 1.
--	unarni	a= --b	a je dodeljena vrednost. b-1 i b je umanjena za 1.
++	unarni	a= b++	a je dodeljena vrednost b i b je uvećana za 1.
--	unarni	a= b--	a je dodeljena vrednost b i b je umanjena za 1.

Unarni operatori

- Operator može da bude ili ispred ili iza promenljive, zavisno od toga da li se **vrednost** promenljive **ažurira pre** ili **posle** izračunavanja izraza.

`y = ++x;` ↔ `x = x+1;
y = x;`

`y = x++;` ↔ `y = x;
x = x+1;`

```
int a, b = 5, c=6;  
a=(b++)+b
```

```
int a, b = 5, c=6;  
a = b++* --c;  
a+=b;
```

Diskutovati

```
int a = 5;
```

```
int b = 4;
```

```
Console.WriteLine(a + b);
```

9

```
Console.WriteLine(a + (b++));
```

9

```
Console.WriteLine(a + b);
```

10

```
Console.WriteLine(a + (++b));
```

11

```
Console.WriteLine(a + b);
```

11

```
Console.WriteLine(14 / a);
```

2

```
Console.WriteLine(14 % a);
```

4

Relacioni operatori dodele

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
==	binarni	a = b == c;	a je dodeljena vrednost true ako je b jednaka c, u suprotnom je false.
!=	binarni	a = b != c;	a je dodeljena vrednost true ako b nije jednaka c, u suprotnom je false.
<	binarni	a = b < c;	a je dodeljena vrednost true ako je b manja od c, u suprotnom je false.
>	binarni	a = b > c;	a je dodeljena vrednost true ako je b veća od c, u suprotnom je false.
<=	binarni	a = b <= c;	a je dodeljena vrednost true ako je b manja ili jednaka c, u suprotnom je false.
>=	binarni	a = b >= c;	a je dodeljena vrednost true ako je b veća ili jednaka c, u suprotnom je false.

a je promenljiva tipa **bool**, dok tipovi promenljivih **b** i **c** mogu biti različiti.

Primer 2 i 3

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

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

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

FALSE

```
bool student;
```

```
string ime="Petar";
```

```
student= ime== "Petar";
```

```
Console.WriteLine(student);
```

TRUE

Logički operatori

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
!	Unarni	$a = !b$	a je dodeljena vrednost true ako je b false, ili vrednost false ako je b true (logičko NE);
&&	Binarni	$a = b \& \& c$	a je dodeljena vrednost true ako b i c imaju vrednost true u suprotnom je false (logičko I);
	Binarni	$a = b c$	a je dodeljena vrednost true ako bar jedna od promenljivih (ili obe) b ili c ima vrednost true u suprotnom je false (logičko ILI);
^	Binarni	$a = b \wedge c$	a je dodeljena vrednost true ako b ili c ima vrednost true, ali ne i ako obe promenljive imaju vrednost true, u suprotnom je false (logičko ekskluzivno ILI).

Primer 5

```
bool a = false;  
bool b = true;  
Console.WriteLine(a && b);  
Console.WriteLine(a || b);  
Console.WriteLine(!b);  
Console.WriteLine(b || true);  
Console.WriteLine((5 != 7) ^ (a == b));
```

Bitovski operatori

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
~	Unarni	a = !b	a je dodeljena vrednost true ako je b false, ili vrednost false ako je b true (logičko NE);
&	Binarni	a = b& c;	a je dodeljena vrednost true ako b i c imaju vrednost true, u suprotnom je false (logičko I);
	Binarni	a = b c	a je dodeljena vrednost true ako b i/ ili c ima vrednost true, u suprotnom je false (logičko ILI);
^	Binarni	a = b ^ c	a je dodeljena vrednost true ako b ili c ima vrednost true, ali ne i ako obe promenljive imaju vrednost true, u suprotnom je false (logičko ekskluzivno ILI);
«	Unarni	a << 1	pomeranje (šiftovanje) bitova ulevo (u primeru za jedno mesto);
»	Unarni	a >> 1	pomeranje (šiftovanje) bitova udesno (u primeru za jedno mesto).

Primer 7

```
byte a = 3; // 0000 0011 = 3
```

```
byte b = 5; // 0000 0101 = 5
```

```
Console.WriteLine(a | b); // 0000 0111 = 7
```

```
Console.WriteLine(a & b); // 0000 0001 = 1
```

```
Console.WriteLine(a ^ b); // 0000 0110 = 6
```

```
Console.WriteLine(~a & b); // 0000 0100 = 4
```

```
Console.WriteLine(a << 1); // 0000 0110 = 6
```

```
Console.WriteLine(a >> 1); // 0000 0011 = 3
```

Prvenstvo dodeljivanja

PRVENSTVO	OPERATOR
NAJVIŠI	. ()
	++, --; +, - (unarni)
	*, /, %
	+ (povezivanje stringova)
	+, -
	<, <=, >=
	==, !=
	&, ^, , <<, >>
	&&
NAJNIŽI	
	?:
	=, *=, /=, %=, +=, -=

Ternarni operator

- Najjednostavniji način da se izvede poređenje jeste pomoću ternarnog (uslovnog) operatora.
- Ternarni operator koristi tri operanda.
- Sintaksa je sledeća:

<test>?<rezultatAkoJeTačno>:<rezultatAkoNijeTačno>

- Ovde se ispituje <test> da bi se dobila logička vrednost, pa je rezultat operator **<rezultatAkoJeTačno>** ili **<rezultatAkoNijeTačno>** u zavisnosti od te vrednosti.
- Ovaj operator je prikladan samo za male zadatke.

Primer

```
int x;
```

```
string uslov;
```

```
Console.WriteLine("Unesi x");
```

```
x = int.Parse(Console.ReadLine());
```

```
uslov = (x < 0) ? "Broj je negativan" : "Broj je  
pozitivan";
```

```
Console.WriteLine(uslov);
```