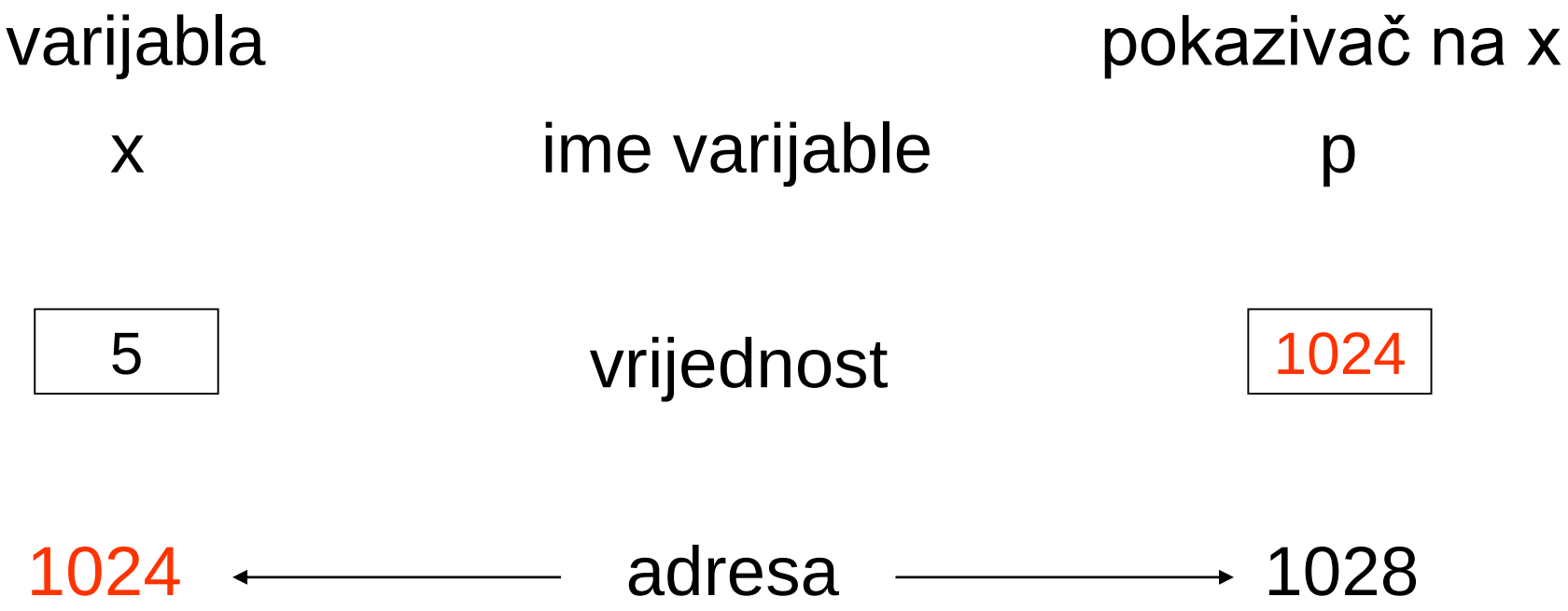
Tipovi varijabli (nastavak)
Operatori i izrazi

Operatori i izrazi

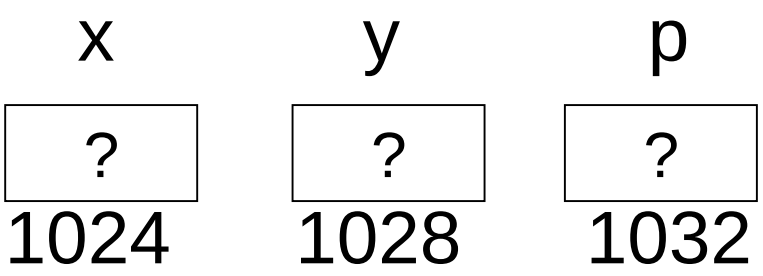
- Nakon odslušanog bit ćete u stanju:
  - opisati osnovne tipove varijabli
  - klasificirati osnovne operatore
  - zapamtiti prioritete
  - izračunati vrijednosti izraza u kojima se koriste operatori.

Pokazivači

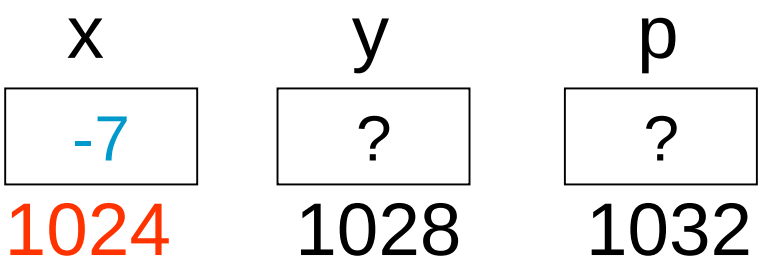

varijabla
pokazivač na x
x
ime varijable
p
5
vrijednost
1024
1024
adresa
1028

Primjer:

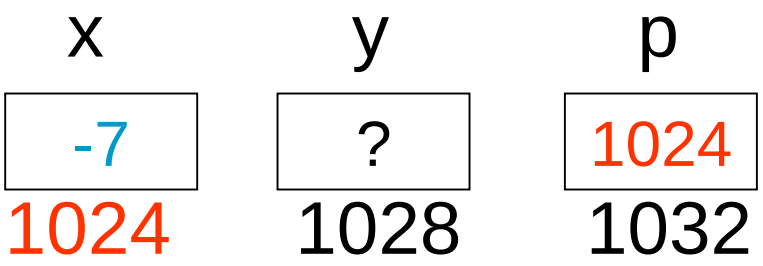
`int` x, y, *p;



x = -7;

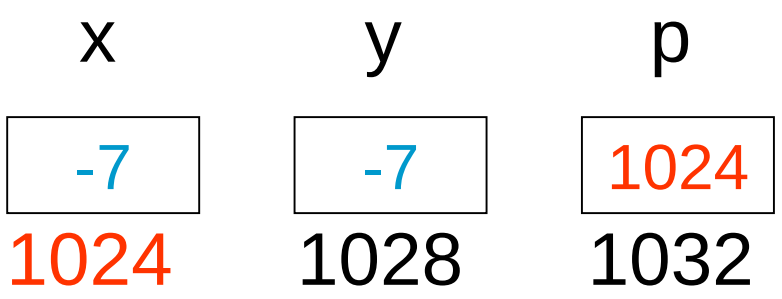


p = &x;

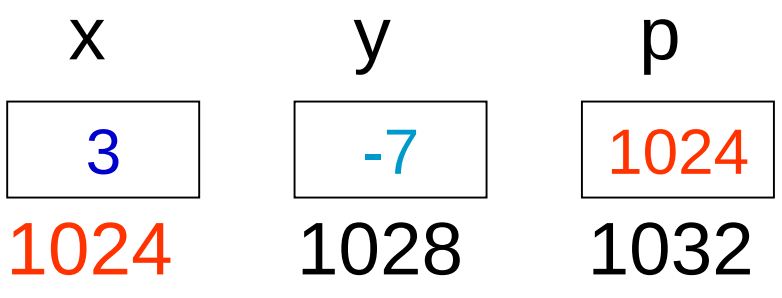


Primjer: (2)

y = *p;



*p = 3;



Enumeracija (pobrojani tip)

```
enum ime{clan_1,clan_2,…,clan_n};
```

Primjer:

`enum` logical {false, true};

`enum` logical flag;

`enum` dan_u_tjednu {po, ut, sr, ce, pe} dan;

`enum` boje {plavo = -1, zuto, crveno, zeleno = 0, sivo};

flag = true;

dan = ce;

Promjena imena nekog tipa

```
typedef stari_tip novi_tip;
```

Primjer:

```
typedef float real;
typedef enum logical boolean;
real x;
boolean flag;
```

Operatori i izrazi - pregled

Operatori:

- pridruživanja
- aritmetički
- relacijski
- logički
- nad bitovima

Izrazi:

- imaju tip i vrijednost
- kombinacija operanada i operatora

Operatori pridruživanja: =

```
varijabla = izraz;
```

Primjer:

x = 3;

y = 3. + 5.342;

a = a + 1;

b = c = 10; /* D→L */

Pretvaranje tipova

- Kada u aritmetičkom izrazu sudjeluju operandi različitih tipova, onda prije izračunavanja izraza dolazi do pretvaranja operanada u najpreciznji tip prisutan u izrazu; rezultat aritmetičkog izraza bit će tog tipa.

- Ako su operandi tipa `float` i `double`, onda će `float` biti pretvoren u `double` prije izračunavanja.

Pretvaranje tipova (2)

- Ako je jedan od operanada realnog tipa (`float`, `double`) a drugi cjelobrojnog (`char`, `short`, `int`, …), onda se cjelobrojni tip konvertira u realni.
- Ako nema realnih operanada, onda se operandi tipa `char` i `short` pretvaraju u tip `int` i zatim se (ako je potrebno) pretvaraju u najpreciziji prisutni cjelobrojni tip.

Aritmetički operatori

Operator	Značenje	Primjer:
-	unarni minus	x = -y;
*	množenje	c = a * b;
/	dijeljenje	a = a/b;
%	modulo	ostatak = a % b;
+	zbrajanje	b = a + b;
-	oduzimanje	x = b - a;

Prioriteti

- unarni: D→L
- multiplikativni (*, /, %): L→D
- aditivni (+, -): L→D
- pridruživanje: D→L

Primjer:

```
int x, y, z;
…
z = x/y * y + x % y;
```

Primjer:

- Koje vrijednosti će se pridružiti cjelobrojnim varijablama a, a1, A i aA nakon izvršavanja sljedećih naredbi:
  - a = 12/5;
  - a1 = 12/5 * 5;
  - A = 12/(5 * 5);
  - aA = 12 % 5;
- Koja vrijednost će se pridružiti cjelobrojnoj varijabli pc nakon izvršavanja naredbe
  - pc = (3 * (8 + 3) - 2)/4;

Operatori inkrementiranja i dekrementiranja

x++ x = x + 1

```
int x, z;
x = 3;
z = x++ - 2;
```

postfix

x-- x = x - 1

```
int x, z;
x = 3;
z = ++x - 2;
```

prefix

Nakon izvršavanja:

z = 1, x = 4

x = 4, z = 2

Operatori inkrementiranja i dekrementiranja (2)

Primjer:

```
i = 7;
printf("i = %d\n", --i);
printf("i = %d\n", i--);
printf("i = %d\n", i);
```

Operatori inkrementiranja i dekrementiranja (3)

Primjer:

```
int a, b;

b = 3;
a = 7;
a = a+++b;

printf("a = %d\n", a);
printf("b = %d\n", b);
```

Relacijski operatori

Operator	Značenje	Primjer
==	jednako	`if`(x == y)…;
!=	različito	`if`(a != b)…;
<	manje	i < 'A' - 'a' + 1
<=	manje ili jednako	x = a + b <= c;
>	veće	c = b > (a + b);
>=	veće ili jednako	a + b >= c

Relacijski operatori (2)

Primjer:

```
int a = 1, b = 20, limit = 100;
int rezultat;

rezultat = a < b;
rezultat = a == b;
rezultat = (a + 10) >= limit;
```

Logički operatori

Operator	Značenje	Primjer
&&	logičko I	(i > 1) && (j < 6)
||	logičko ILI	x = 10 || 20;
!	logička negacija	`if`(!b) brojac = `0`;

Logički operatori (2)

Primjer:

```
int a = 0, b = 10, c = 100, d = 200;
int rezultat;

rezultat = !(c < d);
rezultat = (a - b) && 1;
rezultat = d || b && a;
```

Tablica prioriteta (nepotpuna)

- unarni (!, ++, --, -): D→L
- multiplikativni (*, /, %): L→D
- aditivni (+, -): L→D
- relacijski (<, <=, >, >=): L→D
- relacijski (==, !=): L→D
- logičko I (&&): L→D
- logičko ILI (||): L→D
- pridruživanje (=): D→L

Zadaća:

```
#include <stdio.h>
#define PR(x) printf ("%d\n", (x));

int main(){
   int x, y, z;

   x = - 4 % 4 / 4 + - 4;                PR(x);
   y = 4 / - x ++ - 4;                   PR(x); PR(y);
   y *= z = x + 4 == 4 / -y;             PR(y); PR(z);
   x = x || y && -- z;                   PR(x); PR(y); PR(z);
   PR(++ x && ++ y || ++ z);             PR(x); PR(y); PR(z);
   return 0;
   /* -4, -3, -3, -3, 1, 1, -3, 1, 1, 2, -2, 1 */  }
```

Operatori nad bitovima

Operator	Značenje	Primjer
~	negacija bit-po-bit	b = ~a;
&	logičko I bit-po-bit	x = a & 0x3f77;
|	logičko ILI bit-po-bit	x = a | 0XF401;
^	ekskluzivno log. ILI b-p-b	x = a^ - 1;
<<	lijevi pomak	c = a << 2;
>>	desni pomak	a >>= 2;

Operatori nad bitovima (2)

Primjer:

```
int x = 3, y = 5;
```

~x = -4

x & y = 1

x | y = 7

x ^ y = 6

x << 2 = 12

y >> 2 = 1

Tablica prioriteta (nepotpuna)

- unarni (~, !, ++, --, -): D→L
- multiplikativni (*, /, %): L→D
- aditivni (+, -): L→D
- pomaci (<<, >>): L→D
- relacijski (<, <=, >, >=): L→D
- relacijski (==, !=): L→D
- logičko I bit-po-bit (&): L→D
- ekskluzivno ILI bit-po-bit (^): L→D
- logičko ILI bit-po-bit (|): L→D
- logičko I (&&): L→D

Složeni operatori pridruživanja

Sintaksa :

```
izraz1 op = izraz2;
```

ekvivalentno je s

```
izraz1=izraz1 op izraz2;
```

Primjer:

x += 1; ekvivalentno je s x = x + 1;

i *= j + 1; ekvivalentno je s i = i * (j + 1);

Vrijedi za sljedeće operatore:

+, -, *, /, %, <<, >>, &, |, ~

Uvjetni operator

Sintaksa:

```
izraz1 ? izraz2 : izraz3;
```

Primjer:

predznak = (x < 0) ? –1 : 1;

Prioriteti:

- logičko ILI (||): L→D
- uvjetni operator (?:): D->L
- pridruživanje (=, +=, -=, *=, …): D→L

Operator „zarez”

Sintaksa:

```
izraz1, izraz2;
```

Primjer:

i++, j--;

x = (c = y, c <= 5);

- …
- pridruživanje (=, +=, …): D→L
- zarez (,): L→D