

Konstante i varijable

Slide 1

Skup znakova

- Velika i mala slova engleske abecede: a-z, A-Z.
- Znamenke: 0-9.
- Znakovi:

```
+ - * / = % & #
! ? ^ " ' ~ \ |
< > ( ) [ ] { }
: ; . , _ (bjelina)
```

Slide 2

Komentari. Standardni C-komentar (više linija)

```
/*      Ovo je
      komentar. */
```

Gubitak kóda:

```
/* Ovo je prvi komentar <-- zaboravio zatvoriti
y=3.14;
/* Ovo je drugi komentar. */
```

Komentar unutar komentara (greška):

```
/*
x=2.71; /* Inicijalizacija */
y=3.14;
*/
```

C++ tip komentara (standard C99)

```
x=2.17; // Inicijalizacija
```

Slide 3

Identifikatori

Identifikatori su imena koja se pridružuju različitim elementima programa kao što su varijable polja i funkcije. Sastoje se od slova i brojeva u bilo kojem poretku s ograničenjem da prvi znak mora biti slovo. Velika i mala slova se razlikuju i znak *_* (*underscore*) smatra se slovom.

Slide 4

Primjeri

Primjeri pravilno napisanih identifikatora:

```

x      y13  sum_1  _temp
names  Pov1  table  TABLE
  
```

Primjeri nepravilno napisanih identifikatora:

```

3dan   (prvi znak je broj)
"x"    (nedozvoljeni znak ")
ac-dc  (nedozvoljeni znak -)
  
```

Slide 5

Osnovni tipovi podataka

int: cjelobrojni podatak. Tipično zauzima 4 bajta.

char: znakovni podatak. Sadržava jedan znak iz sustava znakova koji računalo koristi. U memoriji tipično zauzima jedan bajt.

float: broj s pokretnim zarezom u jednostrukoj preciznosti. U memoriji tipično zauzima četiri bajta.

double: broj s pokretnim zarezom u dvostrukoj preciznosti. U memoriji tipično zauzima osam bajtova.

Slide 7

Ključne riječi

auto	extern	sizeof
break	float	static
case	for	struct
char	goto	switch
const	if	typedef
continue	int	union
default	long	unsigned
do	register	void
double	return	volatile
else	short	while
enum	signed	

Slide 6

short, long, unsigned

Cjelobrojni tip `int` može se modificirati pomoću kvalifikatora `short`, `long` i `unsigned`. Tako dobivamo nove cjelobrojne tipove:

```

short int ili short
long int ili long
unsigned int ili unsigned
unsigned short int ili unsigned short
unsigned long int ili unsigned long
  
```

`sizeof(short) <= sizeof(int) <= sizeof(long)`

(`sizeof` daje širinu tipa u bajtovima.)

Slide 8

- Datoteka zaglavlja `<limits.h>` sadrži simboličke konstante koje daju minimalne i maksimalne dozvoljene vrijednosti cjelobrojnih tipova podataka.
- Modifikator `long` može se primijeniti i na realne tipove podataka. Tako je `long float` isto što i `double`, a `long double` je broj s pokretnim zarezom u četverostrukoj preciznosti (tamo gdje takav tip postoji).
- Datoteka zaglavlja `<float.h>` sadrži niz simboličkih konstanti koje daju različite informacije o realnim tipovima podataka.

Slide 9

Varijable istog tipa moguće je deklarirati u istom retku kao u primjeru

```
short a,b,c;
```

ili svaku zasebno

```
short a;
short b;
short c;
```

Slide 11

Deklaracija varijable

```
tip ime;
```

Primjeri:

```
int a,b;
unsigned c;
char d;
```

Slide 10

Deklaracija polja

Polje dimenzije `dim` je niz od `dim` varijabli istog tipa koje su indeksirane cjelobrojnim indeksom koji se kreće u rasponu od 0 do `dim-1`.

Deklaracija polja:

```
tip ime[dim];
```

Primjer:

```
float vektor[10];
```

Elementi polja su

```
vektor[0], vektor[1], vektor[2], .... , vektor[9].
```

Slide 12

Deklaracija pokazivača

Pokazivači su varijable koje sadrže memorijske adrese drugih varijabli.

Deklaracija:

```
tip_p *ime;
```

Primjer:

```
float u;
float *pu;
```

ili

```
float u,*pu;
```

Slide 13

Cjelobrojne konstante

Tipa `unsigned (u)`, `long (l)` i `unsigned long (ul)`

<u>Konstanta</u>	<u>Tip</u>
500000U	<code>unsigned</code> (decimalna)
123456789L	<code>long</code> (decimalna)
123456789ul	<code>unsigned long</code> (decimalna)
0123456l	<code>long</code> (oktalna)
0X50000U	<code>unsigned</code> (heksadecimalna)

Slide 15

Cjelobrojne konstante

Decimalne:

```
0      1      234      -456      99999
```

Oktalne:

```
0      01      -0651      077777
```

Heksadecimalne:

```
0x0      0x1      -0X7FFF      0xabcd
```

Znamenke:

```
a = 10
b = 11
c = 12
d = 13
e = 14
f = 15
```

Slide 14

Znakovne konstante (tipa char)

```
'A'  'x'  '5'  '?'  ' '
'\n'  '\b'  '\\'
```

Znakovne konstante zadane svojim kôdom:

- U obliku `\ooo`, gdje je `ooo` troznamenasti oktalni broj. Npr., `'\170'` = znak koji ima kôd 170 oktarno = 120 decimalno = `'x'` u ASCII znakovima.
- U obliku `\xoo`, gdje `oo` dvoznamenkasti heksadecimalni broj. Npr., `'\x78'` = 120 decimalno = `'x'` u ASCII znakovima.

Slide 16

Posebni znakovi:

Kód	Značenje
<code>\b</code>	povratak za jedno mjesto unazad (backspace)
<code>\f</code>	nova stranica (form feed)
<code>\n</code>	novi red (new line)
<code>\r</code>	povratak na početak linije (carriage return)
<code>\t</code>	horizontalni tabulator
<code>\v</code>	vertikalni tabulator
<code>\0</code>	nul znak (null character)
<code>\?</code>	upitnik
<code>\"</code>	navodnik
<code>\'</code>	jednostruki navodnik
<code>\\</code>	obrnuta kosa crta (backslash)

Slide 17

Konstantni znakovni nizovi (stringovi)

Nizovi znakova navedeni unutar (dvostrukih) navodnika:

```
"Zagreb"    "01/07/2001"
"Linija 1\nLinija 2\nLinija3"
```

Zadnji znak u nizu je nul-znak `'\0'`.

`'a'` nije isto što i `"a"`

Slide 18

Realne konstante

Tipa double:

```
0.    1.    -0.2    5000.
300000. 3e5  3E+5  3.0e+5  .3e6  30E4
```

Tipa float (f) i long double (l):

```
0.00fe-3f  1.345E+8F  -0.2e-4L  5000.0l
```

Slide 19

Simboličke konstante

```
#define ime tekst
```

Preprocesor prije kompiliranja programa svako pojavljivanje simbola `ime` zamijenjuje tekстом `text`. Primjeri:

```
#define PI  3.141593

#define TRUE  1
#define FALSE 0
```

Simboličke konstante olakšavaju razumijevanje programa.

Slide 20

Inicijalizacija varijabli

tip varijabla = konstantni izraz;

Na primjer:

```
int a=7,b;
unsigned c=2345;
char d='\t';
```

Deklaracija konstante pomoću kvalifikatora const:

```
const double e=2.71828182845905;
```

Slide 21

Inicijalizacija polja

Inicijalizacija

```
double x[]={1.2,3.4,-6.1};
```

daje

```
x[0]=1.2, x[1]=3.4, x[2]=-6.1
```

Znakovno polje može se inicijalizirati stringom:

```
char tekst[]="Init";
```

je ekvivalentno s

```
char tekst[]={ 'I', 'n', 'i', 't', '\0' };
```

Slide 22

Enumeracije

Pomoću enumeracija deklariramo simbolička imena koja poprimaju cjelobrojne vrijednosti. Primjer:

```
enum boolean {FALSE, TRUE};
```

Vrijednosti identifikatora:

```
FALSE=0 TRUE=1
```

Deklaracija varijabli:

```
enum boolean x,y;
```

Upotreba:

```
x=FALSE;
.....
if(x==TRUE) y=FALSE;
```

Slide 23

Deklaracija enumeracije

Deklaracija:

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

Vrijednosti identifikatora:

```
clan_1=0
clan_2=1
clan_3=2
.....
clan_n=n-1
```

Deklaracija varijabli:

```
enum ime var_1, var_2, ..., var_m;
```

ili

```
enum ime {clan_1, clan_2, ..., clan_n}
var_1, var_2, ..., var_m;
```

Slide 24

Vrijednosti simboličkih imena

Vrijednosti koje se dodjeljuju identifikatorima mogu se modificirati:

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

Time dobivamo

```
plavo=-1  
zuto=0  
crveno=1  
zeleno=0  
ljubicasto=1  
bijelo=2
```

Slide 25