

Osnovni tipovi podataka

Srednja škola fra Slavka Barbarića Čitluk
Izborna nastava

Mario Miletić

Varijable

- Budući da program obrađuje podatke, upravo se podaci spremaju u memoriju računala.
- Mjesto u memoriji rezervirano za pohranu podatka naziva se **varijabla**.

Memorijske adrese

- Svaka **varijabla** ima jedinstvenu **memorijsku adresu** koja je **višeznamenasti binarni broj**.
- **Korisnicima** je **takav način bilježenja** varijabli **neprikladan** pa se **uvodi označavanje** varijabli **simboličkim imenima**.
- **Simboličko** se **ime naziva** i **identifikator**.

Simboličko ime (identifikator)

- Simboličko ime određuje korisnik poštujući pravila:
 - Smiju se rabiti slova engleske abecede, brojevi i znak _ (podcrtavanje).
 - Broj znakova u simboličkom imenu (dužina) nije ograničen.
 - Mora početi slovom ili znakom _ (podcrtavanje).

Simboličko ime (identifikator)

Napomene za određivanje simboličkog imena:

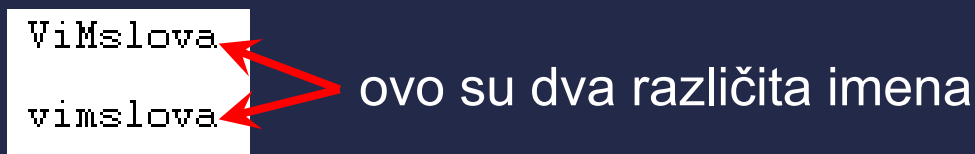
- ne smije se rabiti razmak,
- ne smiju se rabiti naši dijakritički znakovi (č,ć,ž,š,đ),
- ne smiju se rabiti ključne riječi ili oznake operatora programskog jezika.

Simboličko ime (identifikator)

- Program razlikuje velika i mala slova.

```
ViMslowa  
vimslova
```

ovo su dva različita imena



- Ako se koristi ime sastavljeno od više riječi, riječi se mogu odvojiti znakom za podcrtavanje ili pisati spojeno s velikim početnim slovom za svaku

```
ime_sastavljeno_od_vise_rijeci;  
ImeSastavljenoOdViseRijeci;
```

Simboličko ime - primjeri

- Ispravna simbolička imena:

x

promjer_kruga

_kon1

DatumUpisa

Val23m1_X

Simboličko ime - primjeri

■ Neispravna simbolička imena:

1Y (ne smije početi brojem)

x[1] (ne smije sadržavati zagrade)

Datum Upisa (ne smije sadržavati razmak)

goto (ne smije biti ključna riječ)

Brojač1 (ne smije sadržavati naše dijakritičke znakove)

Oznaka tipa podataka

- Svakoj varijabli osim imena treba dodijeliti i oznaku tipa podatka koji će u nju biti smješten.
- Tako će računalo “znati” koliko mjesta u memoriji predvidjeti za pohranu zadanog podatka.

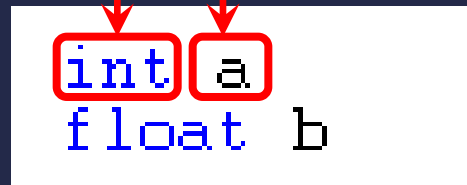
Oznaka tipa podataka

- U programskom jeziku C razlikujemo 5 osnovnih tipova podataka:
 - cjelobrojni – `int`
 - realni – `float`
 - realni dvostruke preciznosti – `double`
 - znakovni – `char`
 - nedefinirana vrijednost - `void`

Deklaracija varijabli

- Postupak pridjeljivanja simboličkog imena varijabli i određivanje tipa podatka naziva se deklariranje.

oznaka tipa podatka *simboličko ime podatka*



```
int a  
float b
```

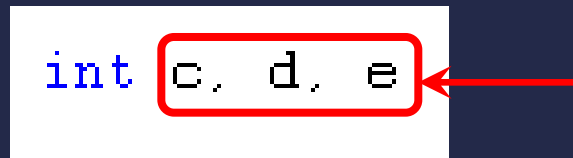
The diagram shows a white rectangular box containing the code 'int a' on the first line and 'float b' on the second line. Above the box, there are two red arrows pointing downwards to the words 'int' and 'a' in the first line. A horizontal red line is positioned above these arrows, with a vertical line extending from its center down to the space between the two arrows.

(Značenje oznaka tipa `int` i `float` objašnjeno je u nastavku.)

Deklaracija varijabli

- Ako je podataka više, odvaja ih se zarezom.

```
int c, d, e
```



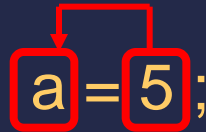
- Gornji se izraz može čitati ovako: “određuje se (deklarira) da su varijable c, d i e tipa int”.

Pridruživanje vrijednosti

- Deklariranoj varijabli se može pridružiti vrijednost.
- Vrijednost joj se pridružuje operatorom pridruživanja.
- Operator pridruživanja je znak =.

Pridruživanje vrijednosti

- Znak = više ne označava izjednačavanje (jednakost) kao u matematici!



a = 5;

- Objektu s lijeve strane operatora pridruživanja pridružuje se vrijednost s njegove desne strane.
- Objekti s lijeve strane operatora pridruživanja moraju biti varijable.

Pridruživanje vrijednosti

- S obzirom na novo značenje znaka jednakosti, u C-u ispravan je i sljedeći izraz:

`a=a+3;`

Treba ga čitati:

“Vrijednost varijable `a` uvećaj za 3”

(Podatku koji se nalazi u varijabli `a` dodaj vrijednost 3 i zatim taj zbroj pohrani u varijablu `a`.)

Pridruživanje vrijednosti

- U istoj se naredbi može koristiti i više operatora pridruživanja (smjer operacije ide s desna na lijevo).

`a=b=c=5;`

Treba čitati:

“Neka varijabla `c` poprimi vrijednost `5`, a varijabla `b` poprimi istu vrijednost koju ima varijabla `c`. Neka varijabla `a` poprimi istu vrijednost koju ima varijabla `b`.”

Cijeli brojevi - int

- Ako je podatak cijeli broj njegova oznaka tipa je *int*.
- Varijabla označena sa *int* je cjelobrojna varijabla.

```
int a
```

```
int c, d, e
```

- Cjelobrojnoj varijabli može se pridijeliti samo cijeli broj.

Cjelobrojne varijable

- Za pohranu cijelog broja u memoriji su predviđena 4 bajta (32 bita).
- Prvi je bit rezerviran za predznak, pa za pohranu broja ostaje 31 bit. (signed int)
- 31 bit omogućava pohranu brojeva iz raspona:
[-2^{31} , $2^{31}-1$] to jest od -2.147.483.648 do 2.147.483.647

Broj bez predznaka

- Sve cjelobrojne varijable mogu biti deklarirane sa ili bez predznaka.
- Ako se deklarira cijeli broj bez predznaka potrebno je ispred oznake tipa staviti ključnu riječ `unsigned`.

```
unsigned int a;
```

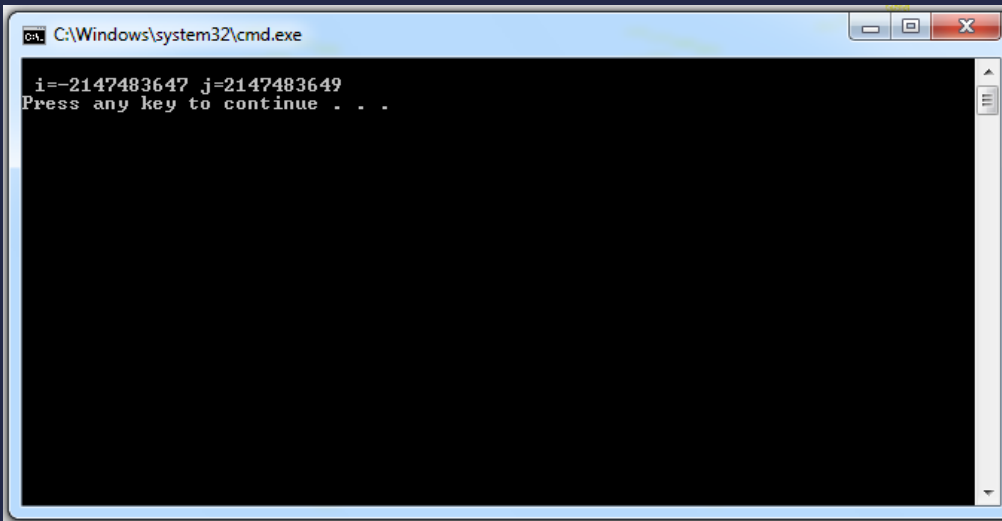
Broj bez predznaka

- U slučaju cijelog broja bez predznaka **bit za predznak više nije potreban.**
- **Najveću je vrijednost** sada moguće prikazati sa 32 bita.
- **Najveći broj** koji se može prikazati sa 32 binarne znamenke je $2^{32} - 1 = 4294967295$.

Primjer 1

```
#include <stdio.h>
void main (void) {
    int i;
    unsigned int j;
    j=2147483649;
    i=j;
    printf ("\n i=%d j=%u \n", i, j);
}
```

← Programski kod koji
unosimo u VS 2010



The screenshot shows a Windows command prompt window with the title bar "cmd - C:\Windows\system32\cmd.exe". The output of the program is displayed on the black background: "i=-2147483647 j=2147483649" followed by "Press any key to continue . . .".

← Rezultat programa

Realni brojevi - float

- Ako je podatak realni broj njegova oznaka tipa je *float*.
- Varijabla označena sa *float* je realna varijabla.

```
float a=4.78;
```

Realni brojevi - float

- Realni brojevi mogu se prikazati:
 - s nepomičnom decimalnom točkom,
 - s pomičnom decimalnom točkom (engl. *floating point*), u eksponencijalnom prikazu.
- C programski jezik za odjeljivanje cjelobrojnog od decimalnog dijela broja rabi decimalnu točku a ne zarez.

Eksponecijalni prikaz broja

- Kada se **realne** brojeve **prikazuje** u **eksponecijalnom prikazu**, s pomičnom decimalnom točkom, oni su oblika:

$$M \cdot 10^E$$

- **M** označava dio broja koji se naziva **mantisa**, a **E** je **eksponent baze 10**.

Eksponencijalni prikaz broja

- Mantisa se zapisuje tako da je prva znamenka različita od nule lijevo od decimalne točke.

$$6.345 = 6.345 \cdot 10^0$$

$$1236.345 = 1.236345 \cdot 10^3$$

$$0.000765 = 7.65 \cdot 10^{-4}$$

Eksponecijalni prikaz broja

- Realni broj se može zapisati:

mantisa ← `9.11e-31` → eksponent

$$6.345 = 6.345 \cdot 10^0 = 6.345e0$$

$$1236.345 = 1.236345 \cdot 10^3 = 1.236345E+3$$

$$0.000765 = 7.65 \cdot 10^{-4} = 7.65e-4$$

Primjer 2

■ Kako računalo pohranjuje broj 6.345?

■ $6.345 = 6 + 0.345$

■ $6_{(10)} = 110_{(2)}$

$6 : 2 = 3$ ostatak 0
 $3 : 2 = 1$ ostatak 1
 $1 : 2 = 0$ ostatak 1

$0.345_{(10)} = 0.010110..._{(2)}$

$0.345 * 2 = 0.69$
 $0.69 * 2 = 1.38$
 $0.38 * 2 = 0.76$
 $0.76 * 2 = 1.52$
 $0.52 * 2 = 1.04$
 $0.04 * 2 = 0.08$

....

$6.345_{(10)} = 110.010110..._{(2)}$

Prikaz realnih brojeva u računalu

predznak		KARAKTERISTIKA (eksponent + 127) (8 bita)						MANTISA bez vodeće jedinice (23 bita)					
31	30	...						23	22	21	...	1	0

$$11.25_{(10)} = 1011.01_{(2)} = 1.01101 * 2^3$$

Predznak: + → 0

Karakteristika : $3 + 127 = 130_{(10)} = 10000010_{(2)}$

Mantisa bez vodeće jedinice : 01101

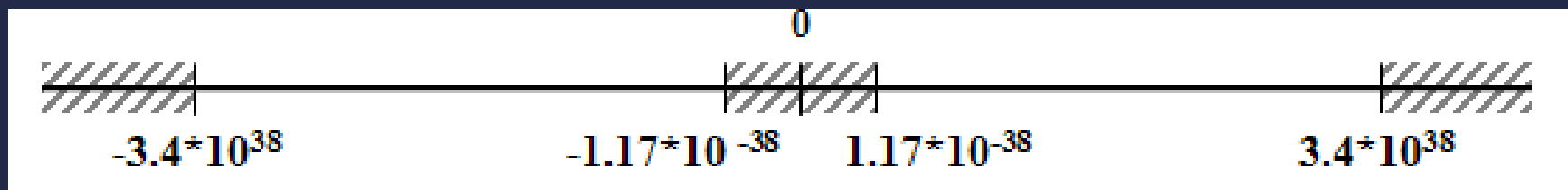
0 | 10000010 | 011010000000000000000000

Realne varijable

- Za pohranu realnog broja u memoriji predviđena su 4 bajta (32 bita).

- Omogućena je pohrana brojeva u rasponu:

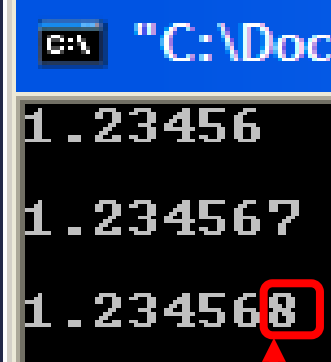
od $[-3.4 \cdot 10^{38}$ do $-1.17 \cdot 10^{-38}]$ do $[1.17 \cdot 10^{-38}$ do $3.4 \cdot 10^{38}]$



Realne varijable

- U realnu se varijablu sprema samo 7 decimalnih znamenki mantise.
- Ako se unese više od sedam znamenki, prilikom prevođenja će biti zanemarene najmanje vrijedne decimalne znamenke (po potrebi se zaokružuje).

```
float a,b,c;  
a=1.23456;  
b=1.23456739;  
c=1.23456789;
```



```
C:\> "C:\Docu  
1.23456  
1.234567  
1.234568
```

Realne varijable veće točnosti

- Ako navedena točnost ne zadovoljava ili ako se žele koristiti brojevi manji od 10^{-38} ili veći od 10^{38} , mogu se rabiti **varijable veće točnosti**.
- To su varijable tipa:
 - *double* (eksponent 308), sa točnošću 15 decimalnih znamenki,
 - *long double* (eksponent 4932) sa točnošću 18 decimalnih znamenki.

Znakovi - char

- Ako je podatak znak, njegova oznaka tipa je *char*.
 - Podatak tipa *char* je predstavljen:
 - jednim znakom unutar jednostrukih navodnika
- ili
- ASCII vrijednošću tog znaka (u dekadskom obliku).

```
char slovoA = 'A';  
char SLOVOA = 65;
```


Znakovi

- Za pohranu znakovnog podatka je u memoriji predviđen 1 bajt (8 bitova).
- Pošto je $2^8 = 256$, moguće je prikazati 256 različitih znakova.
- Znak se pohranjuje kao broj koji predstavlja ASCII vrijednost odabranog znaka.

ASCII kod (prvih 128 znakova)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	00 0000	01 0000	02 0000	03 0000	04 0000	05 0000	06 0000	07 0000	08 0000	09 0000	10 0000	11 0000	12 0000	13 0000	14 0000	15 0000	
	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI	
0	□	▤	⌋	▤	↘	☒	✓	⤴	↵	➤	≡	∇	⇓	⚡	⊗	⊙	8
	16 0001	17 0001	18 0001	19 0001	20 0001	21 0001	22 0001	23 0001	24 0001	25 0001	26 0001	27 0001	28 0001	29 0001	30 0001	31 0001	
	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US	
1	☐	⊖	⊕	⊗	⊘	✓	∩	⊖	⊗	†	‡	⊖	⊕	⊗	⊘	⊙	9
	32 0010	33 0010	34 0010	35 0010	36 0010	37 0010	38 0010	39 0010	40 0010	41 0010	42 0010	43 0010	44 0010	45 0010	46 0010	47 0010	
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	A
	48 0011	49 0011	50 0011	51 0011	52 0011	53 0011	54 0011	55 0011	56 0011	57 0011	58 0011	59 0011	60 0011	61 0011	62 0011	63 0011	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	B
	64 0100	65 0100	66 0100	67 0100	68 0100	69 0100	70 0100	71 0100	72 0100	73 0100	74 0100	75 0100	76 0100	77 0100	78 0100	79 0100	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ō	C
	80 0101	81 0101	82 0101	83 0101	84 0101	85 0101	86 0101	87 0101	88 0101	89 0101	90 0101	91 0101	92 0101	93 0101	94 0101	95 0101	
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	D
	96 0110	97 0110	98 0110	99 0110	100 0110	101 0110	102 0110	103 0110	104 0110	105 0110	106 0110	107 0110	108 0110	109 0110	110 0110	111 0110	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	E
	112 0111	113 0111	114 0111	115 0111	116 0111	117 0111	118 0111	119 0111	120 0111	121 0111	122 0111	123 0111	124 0111	125 0111	126 0111	127 0111	
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	▨	F
																DEL	

Neke ASCII vrijednosti

7	zvučni signal ('\a')
32	praznina (' ')
(48 – 57)	znamenke '0'-'9'
(65 – 90)	velika slova 'A' do 'Z'
(97 – 122)	mala slova 'a' do 'z'

(Nevedene su dekadске vrijednosti ASCII znakova)

Znakovi - char

- Ako je podatak znak koji se ne može prikazati na zaslonu (znakovi iz ASCII tablice od 0-31), koristi se slijed koji počinje lijevom kosom crtom (engl. *backslash*), npr.:

deklaracija	znak	ASCII	značenje
char zvuk	'\a'	7	zvučni signal
char novi_red	'\n'	10	pomak kazala u novi red

Znakovni niz

- Za pohranu teksta se koriste znakovni nizovi (engl. *character strings*).
- Za sada je dovoljno znati da se sadržaj znakovnog niza navodi unutar para dvostrukih navodnika. Npr.:

`“Ovo je znakovni niz”`

Konstante

- U programima se ponekad rabe **simboličke veličine** čija se **vrijednost** tijekom izvođenja programa **ne smije mijenjati**.
- Takve se simboličke veličine nazivaju **konstantama** (npr. fizikalne ili matematičke konstante).

Brojevine konstante

- Kada se u kodu pojavljuju **brojevine konstante**, prevoditelj ih pohranjuje u obliku nekog od osnovnih brojevnih tipova podataka.
- Tako:
 - **realne brojevine konstante** postaju tipa **double**,
 - **cjelobrojne brojevine konstante** postaju tipa **int**.

Brojevne konstante

- Da bi se lakše rukovalo brojevnim konstantama može ih se spremirati u rezervirana mjesta u memoriji.
- Takva rezervirana mjesta treba zaštititi od mogućih neželjenih promjena tijekom odvijanja programa.
- Za zaštitu se koristi ključna riječ *const* koja se dodaje ispred oznake tipa varijable.

```
const double pi = 3.14159265359
```


Konstante

- Ako se u programu **pokuša promijeniti vrijednost konstante**, prilikom prevođenja će program prevoditelj (engl. *compiler*) javiti **pogrešku**.

```
#include<stdio.h>

int main() {
    const double pi=3.14143567;
    pi=pi*2;
    printf("%\lf", pi);
}
```

100 %

Error List

2 Errors 1 Warning 0 Messages

	Description	File	Line
1	error C2166: l-value specifies const object	test.c	6
2	warning C4129: 'l' : unrecognized character escape sequence	test.c	7
3	IntelliSense: expression must be a modifiable lvalue	test.c	5