

PODACI



PODACI - VARIABLE

- Program obrađuje podatke.
- Računalo za svaki podatak predviđa i rezervira mjesto u memoriji.
- Takvo se mjesto u memoriji naziva **varijabla**.

MEMORIJSKE ADRESE

- Svaka varijabla ima **jedinstvenu memorijsku adresu** u obliku višeznamenkastog binarnog broja.
- Takav je način bilježenja neprikladan za korisnike.
- Varijable se stoga označavaju **simboličkim imenima (identifikatorima)**.

SIMBOLIČKO IME (IDENTIFIKATOR)

- Pravila za tvorbu simboličkog imena:
 - Rabe se **slova engleske abecede, brojevi i znak _** (podcrtavanje).
 - Mora početi **slovom** ili znakom **_** (podcrtavanje).
 - **Broj znakova** u simboličkom imenu (dužina) **nije ograničen**.

NEISPRAVNA SIMBOLIČKA IMENA

ime varijable	razlog neispravnosti
suma brojeva	razmak nije dozvoljen
Brojač	ne smiju se rabiti naši dijakritički znakovi (č,ć,ž,š,đ)
goto	ne smiju se rabiti ključne riječi ili oznake operatora programskog jezika
4zbroj	ime ne smije započeti brojem

SIMBOLIČKO IME (IDENTIFIKATOR)

- Program **razlikuje velika i mala slova**.

ViMslova
vinslova

→ ovo su dva različita imena

- Pravilno oblikovano **ime sačinjeno od više riječi**:

vise_rijeci
ViseRijeci

OZNAKA TIPa PODATKA

- Postoje **različite vrste podataka**, npr. cijeli brojevi, realni brojevi, znakovi, nizovi itd.
- Da bi računalo “znalo” **koliko mjesta u memoriji predvidjeti** za pohranu podatka, svakoj varijabli osim imena treba dodijeliti i **oznaku tipa podatka** koji će u nju biti smješten.

DEKLARIRANJE

- Pridjeljivanje **simboličkog imena** varijabli i **označavanje podatka po tipu** naziva se **deklariranje**.

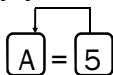
oznaka tipa podatka simboličko ime podatka

```
int a;  
float b;
```

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

PRIDRUŽIVANJE VRIJEDNOSTI

- Deklariranoj varijabli može se **pridružiti vrijednost operatorom pridruživanja**.
- Operator pridruživanja je znak **=**



Znak = više ne označava izjednačavanje kao u matematici!

PRIDRUŽIVANJE VRIJEDNOSTI

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

a=a+3;

- Objektu s lijeve strane operatora pridruživanja pridružuje se vrijednost s njegove desne strane. **Lijevo mora biti varijabla!**

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

TIPOVI PODATAKA

- Podaci se po tipu mogu podijeliti u **osnovne** i **ostale**.
- Osnovni tipovi** su:
 - brojevi** :
 - cijeli brojevi (engl. *integer*),
 - realni brojevi (engl. *floating point*),
 - znakovi**.

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 **cjelobrojnu varijablu** u memoriji su predviđena **4 bajta** (32 bita).
- **Prvi je bit** rezerviran za **predznak**, pa za pohranu **broja** ostaje **31 bit**.
- 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

- **Cjelobrojna varijabla** može biti deklarirana **za pohranu prirodnog broja** (bit predznaka nije potreban):

```
unsigned int a;
```

- Ako bit predznaka više nije potreban, **najveća vrijednost** može se prikazati s **32 bita**.
- Najveći broj je tada $2^{32} - 1 = 4.294.967.295$

REALNI BROJEVI - FLOAT

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

```
float a=4.78;
```

- C++ **za odjeljivanje** cjelobrojnog od decimalnog dijela broja rabi **decimalnu točku**, a ne zarez.

PRIKAZ REALNOG BROJA

- Realni brojevi mogu se prikazati:
 - s **nepomičnom** decimalnom točkom,
 - s **pomičnom** decimalnom točkom (engl. *floating point*), u **eksponencijalnom (znanstvenom) prikazu**.

EKSPONENCIJALNI PRIKAZ BROJA

$M \cdot 10^E$

- **M** je **mantisa**, a **E** je **eksponent** baze 10.
- 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	= $7.65 \cdot 10^{-4}$

EKSPONENCIJALNI PRIKAZ BROJA

- Realni broj se zapisuje:

9.11e-31
mantisa 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

REALNE VARIJABLE

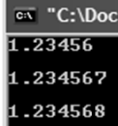
- Za **realnu varijablu** u memoriji predviđena su **4 bajta** (32 bita).
- Omogućena je pohrana brojeva u rasponu:

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

REALNE VARIJABLE

- U realnu se varijablu sprema **samo 7 decimalnih znamenki** mantise.
- Pri unosu **više od 7** znamenki, pri pohrani se **zanemaruju najmanje vrijedne** decimalne znamenke (po potrebi se zaokružuje).

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



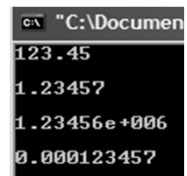
SPREMANJE I PRIKAZ

- Treba uočiti razliku između broja znamenki koje **se spremaju** i broja znamenki koje **se prikazuju na zaslonu!**
- **Broj znamenaka prikaza** na zaslonu se po potrebi **može proširiti** odgovarajućom **naredbom** koja će biti obajšnjena tijekom sljedećih vježbi.

PRIKAZ REALNOG BROJA

- Uobičajeno se realni brojevi prikazuju s **do 6** znamenaka, računajući **od prve različite od 0**.
- Ako se broj ne može prikazati s toliko znamenaka bit će prikazan **u eksponencijalnom prikazu**.

```
float a,b,c,d;
a=123.45;
b=1.234567;
c=1.234560;
d=0.000123456789;
```



REALNE VARIJABLE VEĆE TOČNOSTI

- To su varijable tipa:
 - **double** (eksponent **308**), s točnošću **15** decimalnih znamenki,
 - **long double** (eksponent **4932**) s točnošću **18** decimalnih znamenki.

ZNAKOVI - CHAR

- Ako je podatak znak, njegova oznaka tipa je **char**.
 - Sadržaj se varijabli tipa **char** pridružuje:
 - **znakom unutar jednostrukih navodnika**,
- ili
- ```
char b = 'A';
char c = 65;
```
- upisom **dekadske vrijednosti znaka iz ASCII tablice**.

## ASCII KOD (PRVIH 128 ZNAKOVA)

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9  | A   | B   | C  | D  | E  | F   |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|----|----|----|-----|
| 0 | NUL | SOH | STX | ETX | EOT | ENO | ACK | BEL | BS  | HT | LF  | VT  | FF | CR | SO | SI  |
| 1 | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM | SUB | ESC | FS | GS | RS | US  |
| 2 | SP  | !   | "   | #   | \$  | %   | &   | '   | (   | )  | *   | +   | ,  | -  | .  | /   |
| 3 | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9  | :   | ;   | <  | =  | >  | ?   |
| 4 | @   | A   | B   | C   | D   | E   | F   | G   | H   | I  | J   | K   | L  | M  | N  | O   |
| 5 | P   | Q   | R   | S   | T   | U   | V   | W   | X   | Y  | Z   | [   | \  | ]  | ^  | _   |
| 6 | `   | a   | b   | c   | d   | e   | f   | g   | h   | i  | j   | k   | l  | m  | n  | o   |
| 7 | p   | q   | r   | s   | t   | u   | v   | w   | x   | y  | z   | {   |    | }  | ~  | DEL |

## ZNAKOVI

- Za **znakovnu varijablu** predviđen je **1 bajt**.
- 1 bajt (8 bitova) omogućuje prikazati **256 različitih znakova**.
- Znak se **pohranjuje kao broj** (ASCII vrijednost odabranog znaka).

## 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 |

## NEKE ASCII VRIJEDNOSTI

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

(Nevedene su dekadске vrijednosti ASCII znakova)

## ZNAKOVNI NIZ

- Za pohranu **teksta** (niza znakova) koriste se **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"

## PRIMJER ISPISA ZNAKOVNOG NIZA

```
#include<iostream>
using namespace std;
int main()
{
 cout<<"Prvi niz znakova"<<" "<<"drugi niz znakova.";
 return 0;
}
```

C:\ "C:\Documents and Settings\Sanda\Desktop  
Prvi niz znakova, drugi niz znakova.

## KONSTANTE

- To su veličine čija se **vrijednost tijekom izvođenja** programa **ne smije mijenjati**.  
(npr. fizikalne ili matematičke konstante)
- Može ih se spremiti u rezervirana mjesta u memoriji, ali takva rezervirana mjesta treba **zaštititi od mogućih neželjenih promjena** tijekom odvijanja programa.

## BROJEVNE KONSTANTE

- **Brojevne konstante** pohranjuju se u zadanom tipu podatka:
  - **realne** brojevne **konstante** - tip **double**,
  - **cjelobrojne** brojevne **konstante** - tip **int**.

## KONSTANTE

- Za zaštitu od neželjenih promjena koristi se ključna riječ **const** koja se dodaje **ispred deklaracije**.
- U retku deklaracije konstanti se **pridružuje vrijednost**.

```
const int v_zvuk = 340;

const double pi = 3.14159265359;
```

## KONSTANTE

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

