

Strukture Podataka i Algoritmi

Lekcija 2: Osnovni tipovi podataka

leto 2019/2020

Prof. dr Branimir M. Trenkić

Iz prošle lekcije...

“**Apstraktni tipovi podataka** (ATP) (*Abstract Data Type*, ADT) predstavljaju generalizaciju **osnovnih tipova podataka**, nastali kao rezultat primene postupka apstrakcije u procesu rešavanja problema”

“**Strukture podataka** se grade od **osnovnih tipova podataka** kao gradivnih elemenata, korišćenjem raspoloživih **načina strukturiranja** za njihovo međusobno povezivanje”

Iz prošle lekcije...

“U terminologiji *programskih jezika*, *tip podataka* se obično vezuje za *skup vrednosti* koji neki entitet (promenljiva, konstanta, vraćena vrednost funkcije) može imati”

“*Osnovni ili primitivni* su oni *tipovi podataka* čiji objekti *zadovoljavaju svojstvo atomičnosti* – *skalarni* ne mogu se razdvojiti na prostije celine”

“Takvi tipovi podataka se javljaju kao standardni ili *„ugrađeni“ tipovi podataka* programskog jezika”

C jezik – osnovni tipovi podataka

- Podatak u C-u je numeričkog (broj) ili znakovnog tipa
- **Osnovni tipovi** podataka:
 - **int** (**celobrojna** vrednost),
 - Dekadnim zapisom
 - Oktalnim zapisom (**0**)
 - Heksadecimalnim zapisom (**0x**)
 - Binarni zapisom (**0b**)
 - **float** (razlomljena (**realna**) vrednost),
 - **char** (**znakovna** (karakter) vrednost),
 - **double** (**realna** vrednost **velike tačnosti**).

C jezik – osnovni tipovi podataka

- Bliža specifikacija tipa - **kvalifikatorima**
 - **long** i
 - **short**
 - **Menjaju opseg** dozvoljenih numeričkih vrednosti
- Drugi par kvalifikatora su
 - **signed** i
 - **unsigned**
 - Specificira se da li vrednost može biti **negativna ili ne**
 - Na taj način se, takođe, **bliže specificira dozvoljeni opseg** vrednosti podatka

C jezik – osnovni tipovi podataka

- Napomena: **logički tip podataka** – definisan u mnogim programskim jezicima

$$T := \{v_1, v_2\} \quad v_1 = \bar{F}, \quad v_2 = \bar{T}$$
$$v_1 = 0, \quad v_2 = 1$$

Operacije nad logičkim promenljivama i konstantama : { *not*, *and*, *or* }

C jezik – deklaracija promenljivih

- **Pre** bilo kakve **upotrebe** entiteta u programu
- Pod **deklaracijom promenljive** podrazumeva se ***definisanje skupa vrednosti*** kome mora pripadati podatak koji želimo da smestimo u tu promenljivu
- Deklaracija promenljivih se vrši tzv. **deklarativnim iskazima**

tip_podatka ***ime_promenljive;***

- `int a;` ili, `float val;`
`a = 5;` je dozvoljeno ali `val = 3.72;` je apsolutno dozvoljeno.
`a = 5.0;` nije dozvoljeno.

C jezik – osnovni tipovi podataka

Oznaka	Format	Zahtevana memorija	Opseg
int	%d	2 bajta	-32768 do 32767
float	%f	4 bajta	-3.4e-38 do 3.4e+38
char	%c	1 bajt	-128 do 127
double	%lf	8 bajtova	-1.7e-308 do 1.7e+308

Znakovni tip podataka

$T := \{ '0', '1', \dots, '9', 'A', \dots, 'Z', 'a', \dots, 'z', '+', '-', '=', '/', \dots, \text{kontrolni znaci} \}$

Binarno kodiranje znakova : ANSI - sedmobitni (0 - 127)

$\text{ord}('0') = 48, \dots, \text{ord}('a') = 97;$

$\text{chr}('48') = 0, \dots, \text{chr}('97') = a;$

$'0' < '1' < \dots < '9' < \dots < 'A' < 'B' < \dots < 'Z'$

```
char status;  
status = 'T';
```

Korisnički definisani tipovi podataka

1. Korišćenjem **ključne reči typedef**
2. Kroz **definisanje dozvoljenih vrednosti** novog tipa

Korisnički definisani tipovi podataka

1. Korišćenjem **ključne reči** *typedef*
 - Dodeljivanje **alternativnih imena** postojećim
 - Definisanje **novih (sopstvenih)** tipova podataka

```
typedef int my_int;
```

a zatim možemo deklarirati promenljive

```
my_int x, y, z;
```

što je identično kao i

```
int x, y, z;
```

```
typedef struct {  
    int dan;  
    int mesec;  
    int godina;  
} datum;
```

a zatim možemo deklarirati promenljive

```
datum prodaja, danas;
```

Korisnički definisani tipovi podataka

2. Kroz ***definisanje dozvoljenih vrednosti*** novog tipa

- Ovako definisan tip podataka u C jeziku se naziva ***nabrojiv (enumerisani) tip podataka***

```
enumid name{vrednost};
```

Na primer,

```
enum operator{Sabiranje, Oduzimanje, Mnozenje, Deljenje};
```

```
enum operator opr1, opr2;
```

```
opr1 = Sabiranje;
```

Napomena: enumeracione vrednosti – celobrojne konstante!

Nizovi

- **Jednodimenzionalni niz**
- Definiše se kao **uređena lista** kolekcije elemenata **istog tipa**
- **Sintaksa** kojom definišemo niz,
data_type ime_niza[veličina_niza];

Memorijska reprezentacija:

0	10
1	20
2	40
3	70
4	101
5	30
6	45
7	78
8	75
9	100

Nizovi

- *Pristup određenom elementu* niza
ime_niza[i]
- *i* – *indeks niza* koji određuje redosled elemenata u nizu
 - Donja granica: *0*
 - Gornja granica: *veličina_niza -1*
- Višedimenzionalni nizovi
data_type ime_dvodim_niza[broj_vrsta][broj_kolona];

Pokazivači

- **Pokazivači** nisu ništa drugo do adresa neke memorijske lokacije
- Postoje **dva** važna **operatora** vezanih za pokazivače:
- **Adresni operator**, **&**, sa značenjem „**adresa od**“,
 - primenjuje se za **određivanje adrese promenljive** ili elementa polja
- **Operator indirekcije**, *****, sa značenjem „**vrednost na adresi**“,
 - inverzan je predhodnom, daje **sadržaj** određene memorijske adrese

Pokazivači

- Predpostavimo da postoji sledeća deklaracija celobrojne promenljive ***prom***

int prom = 0;

- Sledi ***deklaracija pokazivača***

int *pok_prom;

- ***Postavljanje pokazivača*** pok_prom na promenljivu prom

pok_prom = &prom;

- Dodeljivanje vrednosti na koju pokazivač ukazuje

x = *pok_prom;

Pokazivačka aritmetika

```
tip *ime1, *ime2;
```

```
ime1=ime2; → ime1 i ime 2 pokazuju na istu memorijsku lokaciju
```

```
*ime1=*ime2; → izjednačavaju se sadržaji memorijskih lokacija
```

a – niz (***a*** – pokazivač na prvi element, ***&a[0], &a[i]***)

v – podatak tipa **T**

pa1 – pokazivač na tip **T**, ukazuje na elemente niza **a**

****pa1*** - vrednost elementa niza **a** na koji ukazuje **pa1**

****pa1 = v*** - dodeljivanje vrednosti elementu

++pa1 - postavlja **pa1** na sledeći element niza

pa1 + n - postavlja **pa1** na **n**-ti naredni element

Strukturalni tip - Definicija

- **Struktura** predstavlja **kolekciju elemenata** u kojoj je **svaki element** identifikovan svojim **vlastitim identifikatorom**
- Svaki element se naziva **članom** (ili **poljem**) strukture
- Struktura je skup više podataka različitih tipova podataka

```
struct {  
    int dan;  
    int mesec;  
    int godina;  
} prodaja, danas;
```

ili

tag

```
struct datum {  
    int dan;  
    int mesec;  
    int godina;  
};  
struct datum prodaja, danas;
```

Strukturalni tip - Definicija

- Alternativni način definisanja strukture pomoću ključne reči ***typedef***

```
typedef struct {  
    int dan;  
    int mesec;  
    int godina;  
} datum;
```

- Na taj način možemo deklarirati različite promenljive tipa strukture koristeći tag datum
datum prodaja, danas;

Strukturalni tip

- **Kako dobiti vrednost članova strukture?**
- **Selekcija članova strukture** razlikuje se od selekcije osnovnih tipova podataka i nizova
- Za **selekciju članova** strukture koristi se **operator . (tačka)**
ime_promenljive.ime_člana
- **Dodela vrednosti** pojedinim polima u strukturi
prodaja.dan = 28;
prodaja.mesec = 6;
prodaja.godina = 2012;

Strukturni tip

- *Kako inicijalizovati strukturu?*
- Inicijalizacija strukture je **slična inicijalizaciji niza**
- *Lista inicijalizacionih vrednosti* se navode između **vitičastih zagrada** razdvojene zarezom

struct datum prodaja = {28, 6, 2012};

Pokazivači i strukture

- Pokazivač na strukturu datum

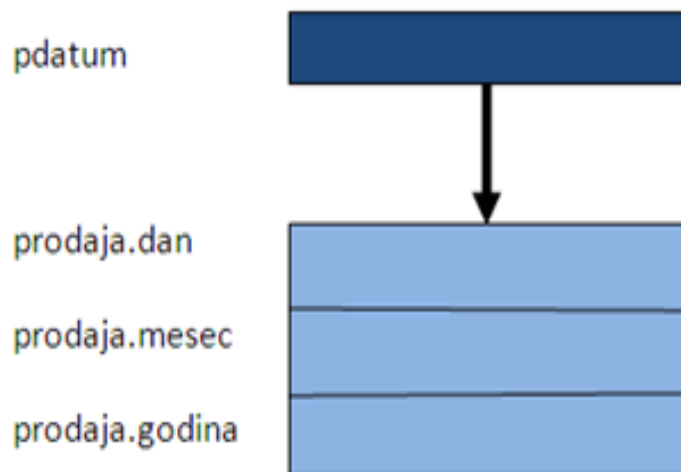
struct datum *pdatum;

- Promenljiva ***pdatum*** je pokazivač na tip struct datum
- ***Postavljanje pokazivača*** na adresu promenljive prodaja koja je tipa strkture datum

pdatum = &prodaja;

Pokazivači i strukture

- **Selekcija člana**
- ***(*pdatum).godina = 2012;***
 - male **zagrade obavezne**
- ***prodaja.godina = 2012;***
ili
- ***pdatum->godina = 2012;***
 - **Bez zagrada** sa operatorom **->** (strelica)



Unije

- Najneobičniji tip podataka u C jeziku
- Koristi se kada je potrebno memorisati **različite tipove podataka u istoj memorijskoj oblasti**
- Na primer, možemo deklarirati promenljivu **x** koja alternativni memoriše karakter, celi ili realan broj

```
union mesoviti_tip {
```

```
    char c;
```

```
    float f;
```

```
    int i;
```

```
};
```

```
union mesoviti_tip x;
```


Složeni tipovi podataka

- ***Nizovi struktura***
 - deklaracija - ***struct datum ndatum[10];***
 - selektovanje elementa - ***ndatum[i].dan***
- ***Nizovi u strukturi***
- ***Strukture u strukturi***

Složeni tipovi podataka

- **Strukture u strukturi – Definisavanje tipa**

```
struct datum_vreme {  
    struct datum date;  
    struct vreme time;  
};
```

- **Deklaracija promenljive:**

```
struct datum_vreme start;
```

- **Selekcija strukture** date

```
start.date
```

Složeni tipovi podataka

- **Strukture u strukturi**
- **Selekcija članova** strukture - višestrukom primenom operatora tačka

start.date.dan = 1;

start.date.mesec = 1;

start.date.godina = 2012;