

Programski jezici i strukture podataka

3

Identifikatori

- Imena:
 - promenljivih,
 - funkcija,
 - makroa,
 - struktura,
 - konstanti,
 - drugi objekti definisani u C prog jeziku.

Pravila za izražavanja identifikatora

- Identifikator se sastoji od:
 - Slova iz osnovnog skupa znakova, uz razlikovanje malih i velikih slova.
 - Donje crte _.
 - Dekadne cifre, prvi znak identifikatora ne sme biti cifra.
- Maksimalan broj karaktera u identifikatoru je po standardu 31, ali u mnogim prevodiocima dužina nije ograničena. Prevodilac zapravo radi sa prvih n značajnih znakova a ostale zanemaruje.

Unapred definisan identifikator

- `main` Imenuje funkciju od koje počinje izvršavanje programa.
- `__func__` (po dve donje crte bez razmaka)
Služi za pristupanje stringu koji sadrži ime funkcije.
- Upotreba prilikom obrade izuzetaka (**z2.c**)

Prostori imena identifikatora

- Imena labela (**z3.c**)
- Tagovi (strukture, unije, nabrojani tipovi)
- Imena članova unija i struktura.
- Ostali identifikatori

Oblasti važenja identifikatora

- file scope (oblast važenja fajl)
- block scope (oblast važenja blok)
- function prototype scope (oblast važenja prototip)
- function scope (oblast važenja funkcija) (**z4.c**)
- Identifikator se može ponovo upotrebiti u oblasti važenja, koja je ugnježdjena u odnosu na njegovu izvornu oblast važenja. (**z5.c**)

Promenljive

- Promenljiva je imenovani objekt koji zauzima memorijski prostor određen tipom.
- Ime promenljive je identifikator.
- Promenljiva pre korištenja treba biti deklarirana.

Tipologija

- Osnovni tipovi (aritmetički tipovi)(skalarni tipovi)
 - celobrojni
 - realni i kompleksni
- Nabrojivi tipovi (aritmetički tipovi)
- Tip void
- Izvedeni tipovi
 - pokazivači (skalarni tipovi)
 - nizovi (agregati)
 - strukture (agregati)
 - unije
 - funkcije

Označeni celobrojni tipovi

- **signed char**
- **int** (signed, signed int)
- **short** (short int, signed short, signed short int)
- **long** (long int, signed long, signed long int)
- **long long** (long long int, signed long long, signed long long int)

Neoznačeni celobrojni tipovi

- `_Bool` (`bool`)
- `unsigned char`
- `unsigned int` (`unsigned`)
- `unsigned short` (`unsigned short int`)
- `unsigned long` (`unsigned long int`)
- `unsigned long long`

Uobičajene veličine pojedinih tipova i opsezi

- **Operator** `sizeof()`
- **datoteka** `limits.h`
- **makroi** `INT_MIN`, `INT_MAX`,
`UINT_MAX`...

Tipovi u formatu sa pokretnim zarezom

- float (4B) preciznost 6 cifara
- double (8B) preciznost 15 cifara
- long double (10B) preciznost 19 cifara

Kompleksni brojevi

- `float _Complex`
- `double _Complex`
- `long double _Complex`
- `complex.h`
- `complex` sinonim za `_Complex`
- makro `I` je imaginarna jedinica (**z6.c**)

Nabrojivi tipovi

- `enum [identifikator] {lista vrednosti};`
- `enum dani {pon, ut, sr, cet, pet, sub, ned};`
- Svaka konstanta u nabrojanju ima svoju vrednost (**z7.c**).

Tip Void

- Nema konstanti i promenljivih tipa `void`
- funkcije tipa `void`
- lista parametara funkcije može biti `void`
- izrazi mogu biti tipa `void`
- pokazivač može biti na tip `void` (**z8.c**)

Logički tip

- U C programskom jeziku tačan je onaj izraz koji ima vrednost različitu od nula, pre svega 1,
- Netačan je onaj izraz koji ima vrednost nula.
- $3==4$ je celobrojni relacioni izraz čija vrednost je netačno.
- Vrednosti tačno i netačno zahtevaju uvođenje posebnog logičkog tipa (`_Bool`). (**z9.c**)

Operacije celog tipa

- Sabiranje, oduzimanje, množenje i deljenje.
- Za obavljanje svake od ovih operacija potrebna su dva cela broja - **operandi**.
- Oznaka operacije se naziva **operator**.
- Kao operatori sabiranja, oduzimanja, množenja i deljenja koriste se znakovi +, -, *, i /, respektivno.

Operacije realnog tipa

- Za realne brojeve su defnisane aritmetičke operacije sabiranje, oduzimanje, množenje i delenje.
- Značenje prva tri operatora (+, -, *) je nepromenjeno.
- Za delenje (/) realnih brojeva važi da je rezultat realan broj.

Konverzija tipova (**cast**)

- Operacija realnog deljenja je predviđena samo za realne brojeve i nije definisana za cele brojeve.
- Da bi podelili dva cela broja, pre deljenja ih pretvaramo u realne brojeve. Ovo se radi, na primer, pri izračunavanju srednje prolazne ocene u okviru pravljenja pregleda ispitnih rezultata.
- Posledica podrazumevajuće konverzije celog tipa u realni je da celi brojevi mogu da budu operandi operatora $/$.

celobrojni aritmetički izraz

$$5/2 \neq 5.0/2.0$$

realni aritmetički izraz

Vrednost prvog izraza 2 se prvo pretvori u realan broj 2. i tek nakon toga se poredi sa vrednošću drugog izraza 2.5 da bi se ustanovilo da je vrednost prethodnog relacionog izraza **tačno**.

Označavanje vrednosti i operacija znakovnog tipa

- Za predstavljanje pojedinačnog znaka koristi se tip **char**.
- Niz znakova zovemo **string**.
- Na primer, izraz 525-864 ako su mu elementi znakovi ne predstavlja aritmetički izraz nego string.

IZRAZI I NAREDBE

leksičke komponente

promenljive
konstante

operandi

+

operatori



izrazi

Svaki izraz ima **tip** i **vrednost**.

Operandi se implicitno konvertuju u odgovarajući tip.

Tip izraza se može eksplicitno promeniti tzv. **cast** operatorom (ispred izraza se u zagradi navede ime tipa u koji želimo da konvertujemo izraz).

(tip)<izraz>

```
int i=3;  
float f=2.4;  
f=i; /* f ce imati vrednost 3.0*/
```

```
int i=3;  
float f=2.4;  
i=f; /* i ce imati vrednost 2*/
```

- Izraz
 - Obavlja računanje i daje rezultat.
 - Uzrokuje neku aktivnost (**bočni efekat**).
- Izuzetak (exception)(**z10.c**)

je stanje programa pri kome određivanje vrednosti nekog izraza daje matematički neodređen rezultat ili rezultat izvan opsega važenja.

- Rezultat aritmetičkih operacija je izraz istog tipa kao i operandi.
- Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip.

Naredba

- Osnovna naredba je izraz iza koga sledi znak ";"
- Imamo više vrsta naredbi:
 - Označena naredba
 - Složena naredba (blok)
 - Naredba izbora
 - Izraz
 - Naredba ponavljanja
 - Naredba skoka

Operatori

- Dodele
- Aritmetički
- Operator sizeof()
- Operatori poređenja
- Logički
- Operatori nad bitima
- Inkrementacija i dekrementacija
- Pomeranje
- Adresa -od i posredni pristup

Operatori

- **Unarni operatori** - prefiksni i sufiksni.
- **Binarni operatori** - infiksni.

Prioritet i asocijativnost operatora

- Unarni operatori imaju viši prioritet od binarnih.
- Aritmetički operatori su višeg prioriteta od relacionih, a ovi višeg od logičkih.
- Operatori dodele su najnižeg prioriteta.
- Ako dva operatora imaju isti prioritet, onda se u obzir uzima asocijativnost, koja može biti s leva na desno, ili s desna na levo.
- Prioritet operatora se može promeniti korišćenjem zagrada ().

Deljenje - konverzije

Kako izbeći celobrojno deljenje

```
int a,b;
```

```
float c;
```

```
a = 5;
```

```
b = 2;
```

```
c = a/b; /* Celobrojno deljenje, c=2*/
```

```
c = (1.0*a)/b; /*Implicitna konverzija: 1.0*a je realan
```

```
broj pa prilikom deljenja sa b dobija se realan rezultat c=2.5*/
```

```
c = (0.0+a)/b; /* Implicitna konverzija: (0.0+a) je realan broj pa prilikom  
deljenja sa b dobija se
```

```
realan rezultat c=2.5*/
```

```
c = (float)a/(float)b; /* Eksplicitna konverzija*/
```

Operatori sa bočnim efektima

- Pojava da se prilikom izračunavanja nekog izraza menja vrednost neke promenljive naziva se **bočni efekat** (eng. side effect).
- Operatori koji imaju bočni efekat su operatori dodele i operatori uvećanja i umanjenja.
- Naredba dodele se u C-u predstavlja izrazom dodele, a njeno izvršavanje se zasniva na bočnim efektima operatora dodele.

Operatori dodele =

- Levi operand ovog operatora je leva vrednost (ime promenljive). Desni operand je proizvoljni izraz.
- Najpre se izračunava izraz na desnoj strani. Njegova vrednost se zatim po potrebi konvertuje u tip promenljive na levoj strani. Nakon toga se dobijena vrednost upisuje u memorijsku lokaciju koja je rezervisana za čuvanje te promenljive.

Operatori umanjenja i uvećanja

- Unarni operatori **++** i **--** nazivaju se operatori **uvećanja** (inkrementacije) i **umanjenja** (dekrementacije) respektivno.
- Mogu biti prefiksni i sufiksni.
- U oba slučaja vrši se uvećanje promenljive za 1 ali izraz **++n** uvećava promenljivu *n* *pre* nego što se njena vrednost koristi, dok **n++** uvećava *n* *nakon* što se njena vrednost koristi. Tako se `x=++n;` razlikuje od `x=n++;`.
- Slično za operator **--**, s tim što je u pitanju umanjenje za jedan.

(z11.c)