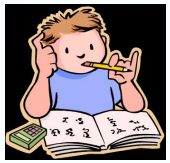
Programski jezik C

Tomaž Dobravec

Strukture



Naloga 5-IX. (tipi/struct.c)

Napiši program, ki prebere ime, priimek in starost petih oseb.

... (najmanj) tri možne rešitve

... najelegantnejša: uporaba struktur

Strukture

Deklaracija strukture

```
struct oseba {
  char ime[10];
  char priimek[20];
  int  starost;
};
```

Uporaba:

```
  struct oseba o;

  scanf("%s", o.ime);
  scanf("%s", o.priimek);
  scanf("%d", &o.starost);

  printf("%s, %s, %d", o.ime, o.priimek, o.starost);
```

Rezervirana beseda `typedef`

```
typedef int stevilo; // stevilo postane sinonim za int
stevilo a;           // spremenljivka tipa int

// kazalci
typedef char *niz;
niz s1,s2;

// tabele
typedef char kratkaTabela[10], dolgaTabela[100];
kratkaTabela kt;  // tabela znakov velikosti 10
dolgaTabela  dt;  // tabela znakov velikosti 100

// strukture
struct tocka {   // struktura 'tocka' z dvema koord.
  int x, y;
};
typedef struct tocka t; // nov tip (t)
t A, B;                 // tocki A in B

// strukture (krajse)
typedef struct kompleksno { // uvedba tipa cplx
  float re, im;
} cplx;
cplx w, z;  // kompleksni stevili w in z
```

6. Spremenljivke

Osnovni podatki o spremenljivki:

podatek	pomen
ime	do spremenljivke dostopamo preko njenega imena
tip	kakšne podatke lahko spremenljivka hrani
vidljivost	kje je spremenljivka vidna
naslov	naslov v pomnilniku, kjer je rezerviran prostor za spremenljivko
velikost	količina pomnilnika rezervirana za spremenljivko
vrednost	trenutna vrednost spremenljivke

Primer:

```
void blabla() {
  int x;
}
```

Globalne in lokalne spremenljivke

Globalna spremenljivka je deklarirana izven vseh funkcij

```
// globalna spremenljivka
int stevilo=0;

// povecam vrednost spremenljivke stevilo
void nov() {
  stevilo++;
}
```

Lokalna spremenljivka je deklarirana v funkciji

```
void blabla() {
  int x;
}
```

Vidljivost

- Bločna vidljivost: spremenljivka je vidna le znotraj bloka, v katerem je deklarirana

- Programska vidljivost: globalne spremenljivke so vidne v vseh funkcijah programa

- Datotečna vidljivost: globalne spremenljivke lahko ”skrijemo” z uporabo rezervirane besede `static`

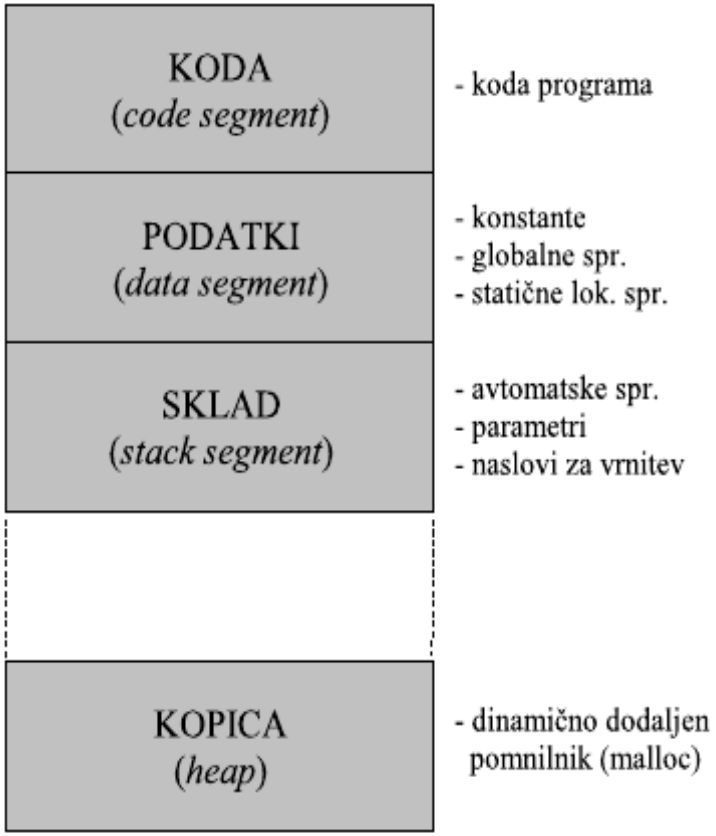
```
static int vrednost;
```

Statične lokalne spremenljivke

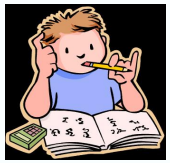
```c
#include <stdio.h>

void bla() {
  static int prvic=1;
  if (prvic==1) {
    printf("Prvi klic funkcije bla()\n");
    prvic = 0;
  } else {
    printf("Naslednji klic funkcije bla()\n");
  }
}

main() {
  bla();
  bla();
}
```

Organizacija pomnilnika po segmentih

Segment	Opis
KODA (*code segment*)	koda programa, prevedena v zbirni jezik
PODATKI (*data segment*)	globalni podatki
SKLAD (*stack segment*)	lokalne spremenljivke in drugi začasni podatki



Primer napačne uporabe tabele



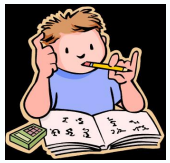
Naloga 6-IIa.

(razno/ciklanje.c)

Kaj se zgodi, ko poženem spodnji program?

```
#include <unistd.h>

int main() {
  int i;
  int a[10];
  for(i=0; i<=10; i++) {
    printf("%d\n", i);
    sleep(1);
    a[i]=0;
  }
}
```

Organizacija pomnilnika



Naloga 6-II.

V katerem delu pomnilnika je rezerviran prostor za posamezne dele spodnjega programa?

```
1  const char msg[] = "Napaka!";
2  int stevec=0;
3
4  void bla(int a) {
5
6    static int i=0;
7  }
8
9  main() {
10   int  x;
11   bla(5);
12 }
```

Program za vajo *(razno/spomin.c)*

Inicializacija spremenljivk

```
int x;          // avtomatska inicializacija (x=0)

static int y; // avtomatska inicializacija (y=0)

int t[100];     // avtomatska inicializacija
                // (v vseh poljih tabele t so nicle)

void bla() {
  static int i;  // avtomatska inicializacija (i=0)

  int j;         // NI avtomatske inicializacije
                 // (v j je lahko karkoli)

  int a[10];     // NI avtomatske inicializacije
                 // (v poljih tabele a je lahko karkoli)

  int *p;

  // NI avtomatske inicializacije
  // v delu pomnilnika, ki ga rezerviramo s funkc. malloc
  // (torej v poljih p[0], p[1], ...), je lahko karkoli
  p = (int *) malloc (100 * sizeof(int));
}
```