Workshop 3

Strukture (prvi dio)

U ovom poglavlju će biti opisane strukture u C++ programskom jeziku. Razumijevanje struktura je bitno za lakši kasniji rad sa klasama (koje su vrlo slične strukturama). Iako strukture postoje i C programskom jeziku, C++ uvodi neke novine i olakšanja pri radu sa njima. Ovdje će biti opisane strukture na C++ način.

Razlika između *strukture* i *objekta*

Struktura predstavlja tip podatka. Strukturu kreira programer, pa možemo reći da je to korisničko definisan tip podatka. Sa naredbom `int a` se deklariše varijabla "`a`" koji je tipa *int*, a sa naredbom `float b` se podatak tipa *float*. Kao što su ovdje *int* i *float* tipovi podataka, tako je struktura je tip podatka (mada je to složeniji tip podatka, kao što ćete kasnije vidjeti). A ono što *int* predstavlja za `a`, i što *float* predstavlja za `b`, to struktura predstavlja za neki objekat.

Strukturu najčešće deklarišemo kao globalnu da bi je mogli koristiti u čitavom programu (u glavnoj funkciji `main`, kao i svim ostalim funkcijama). Da bi struktura bila globalni tip podatka moramo je deklarisati iznad funkcije `main`. Naredni primjer pokazuje deklaraciju strukture `student`. Mi ćemo tu strukturu pojednostaviti, koristi ćemo samo minimalne osobine koje opisuju nekog *studenta – ime*, *broj indeksa*, *godina rođenja*, *godina studija*.

```
1:   #include <iostream>
2:   using namespace std;
3:
4:   struct student
5:   {
6:         char ime[10];              // ime studenta je niz karaktera duzine 10
7:         int indeks;                // broj indeksa je obični integer
8:         int g_rodjena;             // godina rođenja je obični integer
9:         int godina_studija;        // godina studija je obični integer
10:  };
11:
12:  void main()
13:  { ...
```

- nakon ključne riječi `struct` (linija 3) slijedi neko ime strukture npr. student, pacijent, krug
- nakon `struct` *`NekoIme`* slijedi tijelo strukture koje se nalazi između vitičastih zagrada linija 5 i 10, a potom se klasa završava sa znakom tačka-zarez;
- ova struktura sadrži 4 obilježja, njihova deklaracija je ista kao što smo to do sad radili linija 6, 7, 8 i 9

Kako vidite deklaracije strukture je veoma jednostavna. Za vježbu deklarišite strukturu pacijent i dodajte neka jednostavna obilježja.

Slijedeći primjer pokazuje deklaraciju strukture `kruznica`. Prije nego što pogledate primjer razmislite o sljedećem pitanju:

Kada želite nacrtati kružnicu u koordinatnom sistemu, koje osobine kružnice morate poznavati?

Na postavljeno pitanje postoje dva odgovora u zavisnosti kako ćete posmatrati tačku u koodinatnom sistemu.

Primjer pomoću tačke kao cjeline	**Primjer sa prikazanim atributima tačke** (tačka je razložena na njene atribute)
1. tačka O (centar) 2. veličina poluprečnika kružnice	1. `x`-pozicija tačke O (centra) 2. `y`-pozicija tačke O (centra) 3. veličina poluprečnika kružnice

S obzirom, da ne postoji podatak tipa tačka u C++-u, koristit ćemo desni primjer za kreiranje strukture. Na osnovu ova tri obilježja iz desnog primjera kreirat ćemo strukturu `kruznica`:

```
1:   struct kruznica
2:   {
3:       float x;
4:       float y;
5:       float poluprecik;
6:   };
```

Slijedeći primjer pokazuje deklaraciju strukture `trougao`. Prije nego što pogledate primjer razmislite koji su nam podaci potrebi da bi opisali neki trougao u koordinatnom sistemu.

Trougao u koordinatnom sistemu je potpuno definisan sa 3 tačke (A, B i C), pa će nam biti potrebno 6 obilježja a to su `x` i `y` koordinate za svaku tačku trougla.

```
1:   struct trougao
2:   {
3:       float a_x;
4:       float a_y;
5:       float b_x;
6:       float b_y;
7:       float c_x;
8:       float c_y;
9:   };
```

Kreiranje objekata

Sa naredbom `int a;` (donji kôd - linija br. 3) deklarisali smo varijablu `a` koji je tipa *int*, a tako i varijablu `b` tipa *float*. Kao što znamo u ovim varijablama možemo čuvati vrijednosti kao što su `5`, `7`, `1236`, … odnosno `8.56`, `7.78`, odnosno informacije o numeričkoj vrijednosti. Zamislimo da imamo „varijablu" student, u kojoj možemo čuvati sve informacije o jednom studentu. Upravo to možemo postići pomoću strukture i objekata. U strukturi definišemo koje sve informacije treba da sadrži ta „varijablu" (ime, prezime, indeks), a na osnovu te strukture kreiramo našu „varijablu". U liniji 5, dakle, definišemo, „varijablu" `c`, a u liniji 6 „varijablu" `d`. Možemo reći da su ,,varijable'' `c` i `d` tipa student.

Stručni naziv za „varijablu" čiji je tip neka struktura nazivamo „objektom" ili „instancom". Od sad ćemo koristiti termin „objekat" ili „instanca strukture".

```
1:   void main()
2:   {
3:       int a;
4:       float b;
5:       student c;
6:       student d;
```

Naravno, deklaraciju objekata u linijama 5 i 6 možemo deklarisati u jednoj liniji koda:

```
6:       student c, d;
```

Pitanje br. 1:

Za što u deklaraciji strukture ne smijemo i ne možemo inicijalizovati obilježja te strukture, npr. kod deklaracije strukture `student`, koji smo već definisali, liniju br. 7 zamijenimo linijom:

```
7:       int indeks = 1000;
```

Odgovor br. 1?

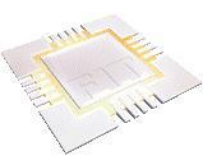
Zato što struktura sa svojim obilježjima predstavlja tip podatka. Dodjela vrijednosti tipu podatka izgleda kao da neko pokuša slijedeće:

```
0:       float = 3.14;  // nema smisla
```

Evo drugi razlog zašto prethodno nema smisla:

Čiji bi indeks imao vrijednost 1000? Student `c` ili student `d`?

U toku vašeg daljeg učenja o strukturama kroz zadatke, sami ćete sebi naći još razloga zašto prethodno nema smisla.



Pitanje br. 2:

Koje je naredno pitanje ispravno postavljeno, a u jednino je to i pitanje br. 3 ?

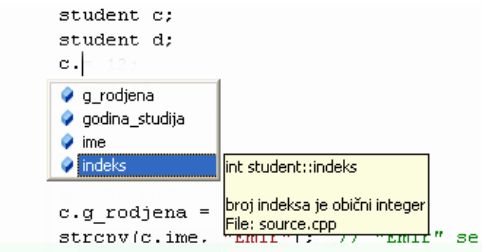
Kako pristupiti (npr. ispisati) vrijednost nekog obilježja (npr. indeks) neke strukture?
Kako pristupiti (npr. ispisati) vrijednost nekog obilježja (npr. indeks) nekog objekta?

Odgovor br. 3:

Pogledajte sljedeći kôd.

```
1:   #include <iostream>
2:   using namespace std;
3:
4:   struct student
5:   {
6:         char ime[10];       // ime studenta je niz karaktera duzine 10
7:         int indeks;         // broj indeksa je obični integer
8:         int g_rodjena;      // godina rođenja je obični integer
9:         int godina_studija; // godina studija je obični integer
10:  };
11:
12:  void main()
13:  {
14:        int a;
15:        float b;
16:        student c;
17:        student d;
18:        a = 12;
19:        b = 3.14;
```

U linijama 14 i 15 pristupamo vrijednostima varijabli `a` i `b`. Kad bi htjeli pristupiti objektu `c`, ne bi smjeli koristiti naredbu `c = 100`, jer kompajler ne zna kojem obilježju treba dodijeliti vrijednost 100. Pristup se vrši tako što nakon naziva objekta stavljamo operator tačku, a zatim napišemo ili odaberemo naziv nekog obilježja npr. indeks :



Dovršite prethodni program:

```
20:        c.indeks = 931;
21:        c.g_rodjena = 1982;
22:        strcpy(c.ime, "Emir");
23:        c.godina_studija = 3;
24:
25:        d.indeks = 811;
26:        d.g_rodjena = 1984;
27:        strcpy(d.ime, "Adil");
28:        d.godina_studija = 3;
29:
30:        cout << "Ja se zovem " << c.ime;
31:        cout << ", imam " << 2006 - c.g_rodjena << " godina! \n";
```

```
/* ova (nova) funkcija strcpy
kopira string "Emir" u niz c.ime
*/
```

```
32:
33:         cout << "On se zove " << d.ime;
34:         cout << "i ima " << 2006 - d.g_rodjena << " godina! \n";
35:  }
```

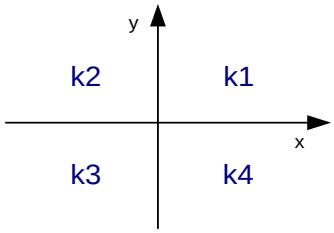
U linijama 20 do 23 pristupate objektu `c`, a u linijama 25 do 28 pristupate objektu `d` koji su tipa `student`, dodjeljujući obilježjima vrijednosti.

Možete zaključiti da sam naziv objekta (npr. `c`) u kodu ne znači mnogo, nego se neki objekat koristi najčešće uz kombinaciju nekih od svojih obilježja, tako da npr. `c.indeks` predstavlja *integer*, `c.g_rodjenja` također *integer*, `c.godina_studija` predstavlja *integer*, a `c.ime` predstavlja neki niz karaktera (`char`) dužine 10.

Zadatak 67:

Pomoću strukture `kruznica` koju smo već deklarisali prije, kreirajte objekte `k1`, `k2`, `k3` i `k4`. Dodijelite vrijednosti obilježjima tako:

a) da se centar kružnice `k1` nalaziti u prvom kvadrantu koordinatnog sistema, `k2` u drugom, `k3` u trećem, a `k4` u četvrtom kvadrantu (na desnoj slici su označena sva četiri kvadranta)



b) sa tastature učitajte vrijednost poluprečnika kružnice `k1`

c) neka poluprečnik kružnice `k2` bude iste veličine kao kod `k1`

d) neka poluprečnik kružnice `k3` bude duplo veći od `k2`

e) neka poluprečnik kružnice `k4` bude zbir poluprečnika od `k2` i `k3`

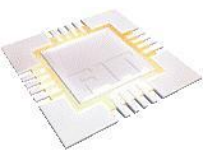
A zatim, ispišite karakteristike kružnica (obilježja objekata k1, …,k4) koristeći pune rečenice, kao u sljedećem primjeru:

```
-  Kružnica iz drugog kvadranta sa centrom O(3,4) ima isti poluprečnik kao
   kružnica O(-4,6) iz prvog kvadranta koji iznosi r = 2.

-  Kružnica O(-9,-20) iz trećeg kvadranta ima duplo veći
-  poluprečnik (r = 4) od kružnice O(-4,6).

-  Kružnica O(11,-23) ima poluprečnik (r = 6) koji predstavlja zbir
   poluprečnika kružnica O(-9,-20) i O(11,-23)
```

Rješenje se nalazi na stranici 44.



Zadatak 68:
Koristeći prethodnu strukturu `trougao` i `kruznica`:

a. kreirajte objekat `T1` (ΔABC) čije ćete koordinate tačaka `A`, `B` i `C` učitati sa tastature
b. kreirajte kružnicu `K1`, čiji se centar nalazi u tački `A` trougla `T1` (ΔABC)
c. vrijednost poluprečnika kružnice `K1` učitajte sa tastature
d. kreirajte kružnicu `K3` sa istim osobinama kao kružnica `K1`, tj. `K1` iskopirajte u novi `K3`
e. povećajte poluprečnik kružnice `K3` za 10%
f. kreirajte kružnicu `K2` koja se nalazi na tački `B` sa istim poluprečnikom kao kod `K3`

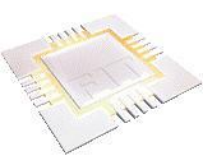
Rješenje se nalazi na stranici 44.

Zadatak 69:
Napravite funkciju `void infoKruznica`, čiji je ulazni parametar podatak tipa `kruznica`. Funkcija treba ispisati osobine kružnice u sljedećem obliku:

```
## infoKruznica
   centar O(1.4, 2.5)
   poluprečnik r = 12
```

Zatim, iskoristite ovu funkciju da bi ispisali osobine kružnice `K1` i `K2` iz zadatka br. 2.

Rješenje se nalazi na stranici 45.



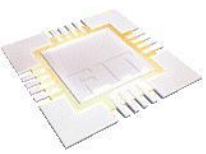
Rješenja

Rješenje zadatka br. **67**:

```
1:   #include <iostream>
2:   using namespace std;
3:
4:   struct kruznica
5:   {
6:          float x;
7:          float y;
8:          float poluprecik;
9:   };
10:
11:  void main()
12:  {
13:         kruznica k1, k2, k3, k4;
14:
15:         k1.x = 3;
16:         k1.y = 4;
17:         cout << "Unesi r1: ";
18:         cin >> k1.poluprecik;
19:
20:         k2.x = -4;
21:         k2.y = 6;
22:         k2.poluprecik = k1.poluprecik;
23:
24:         k3.x = -9;
25:         k3.y = -20;
26:         k3.poluprecik = k1.poluprecik * 2;
27:
28:         k4.x = 11;
29:         k4.y = -23;
30:         k4.poluprecik = k1.poluprecik + k2.poluprecik;
31:
32:         cout << "Kružnica iz drugog kvadranta sa centrom O(";
33:         cout << k1.x << "," << k1.y << ") ima isti poluprečnik kao \nkružnica O(";
34:         cout << k2.x << "," << k2.y << ") iz prvog kvadranta koji iznosi r = ";
35:         cout << k1.poluprecik ;
36:         //....
37:  }
```

Rješenje zadatka br. **68**:

```
1:   #include <iostream>
2:   using namespace std;
3:
4:   struct kruznica
5:   {
6:          float x;
7:          float y;
8:          float poluprecik;
9:   };
10:
11:  struct trougao
12:  {
13:         float a_x;
14:         float a_y;
15:         float b_x;
16:         float b_y;
```



```
17:         float c_x;
18:         float c_y;
19:  };
20:
21:  void main()
22:  {
23:         trougao T1;
24:
25:         cout << "unesi x1 i y1 za tacku A \n";
26:         cin >> T1.a_x >> T1.a_y;
27:
28:         cout << "unesi x2 i y2 za tacku B \n";
29:         cin >> T1.b_x >> T1.b_y;
30:
31:         cout << "unesi x3 i y3 za tacku C \n";
32:         cin >> T1.c_x >> T1.c_y;
33:
34:         kruznica K1;
35:
36:         K1.x = T1.a_x;
37:         K1.y = T1.a_y;
38:
39:         cout << "Unesite r za K1 \n";
40:         cin >> K1.poluprecik;
41:
42:         kruznica K3;
43:
44:         // kopiranje cijelog objekta:
45:         K3 = K1;
46:
47:         /*****************************
48:         Kopiranje svih obiljezja iz K1 u K3 (umjesto linije br. 46)
49:         K3.x = K1.x;
50:         K3.y = K1.y;
51:         K3.poluprecik = K1.poluprecik;
52:         *****************************/
53:
54:         K3.poluprecik = K3.poluprecik * 1.1;
55:
56:         kruznica K2;
57:         K2.x = T1.b_x;
58:         K2.y = T1.b_y;
59:         K2.poluprecik = K3.poluprecik;
62:  }
```

Rješenje zadatka br. **69** (zadatak se nastavlja, obratite pažnju na brojeve linija kôda):

```
     //...
21:  void infoKruznica(kruznica); // prototip funkcije
     //...
60:         infoKruznica(K1);
61:         infoKruznica(K2);
62:  }
63:
64:  void infoKruznica(kruznica u1)
65:  {
66:         cout << "\n## infoKruznica \n centar O("
67:              << u1.x << "," << u1.y << ")    \n poluprecnik r = "
68:              << u1.poluprecik << endl << endl;
69:  }
```