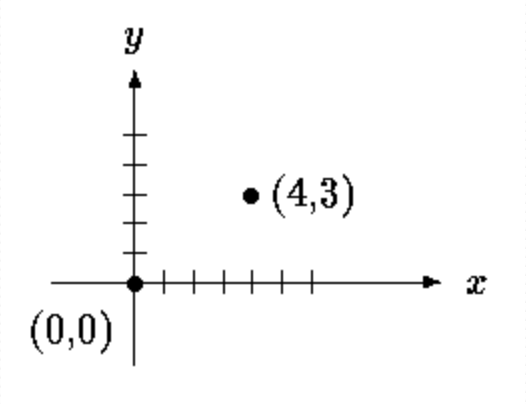
Programiranje i programski jezici

Osnove struktura




```
struct point
{
  int x;
  int y;
};
```

Strukture se deklarišu pomoću ključne reči `struct`

Naziv strukture može se pisati iza reči `struct`

Promenljive članice strukture se navoda između vitičastih zagrada

Promenljive članice mogu imati isti naziv kao i obične promenljive ili kao članice neke druge strukture

Osnove struktura

Deklaracija `struct` definiše tip
Iza deklaracije strukture može se nalaziti lista promenljivih tog tipa

```
struct { ... } x, y, z;
```

potpuno analogno kao

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

Deklaracije strukture iza koje se ne nalazi lista promenljivih ne izaziva rezervisanje memorije. Ona predstavlja šablon strukture.
Naziv neke strukture se može iskoristiti za deklarisanje primeraka te strukture
Ako je prethodno deklarisana struktura *point*, onda

```
struct point pt;
```

deklariše promenljivu *pt* koja je struktura tipa struct *point*
Struktura se može inicijalizovati navođenjem inicijalnih vrednosti

```
struct point maxpt = { 320, 200 };
```

Član određene strukture se može upotrebiti korišćenjem oblika

ime_strukture.clan

Na primer, koordinate tačke pt možemo prikazati pomoću

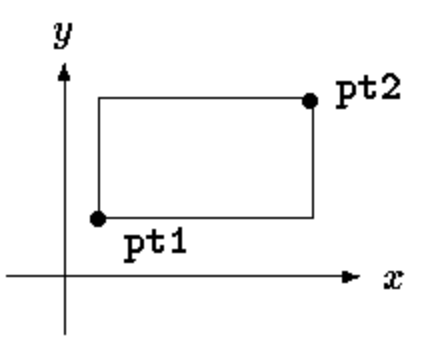
```
printf("%d,%d", pt.x, pt.y);
```

Slično, rastojanje od koordinatnog početka do tačke pt možemo izračunati kao

```
dist = sqrt((double)pt.x * pt.x + (double)pt.y * pt.y);
```

Članice struktura mogu biti takođe strukture

```
struct rect
{
   struct point pt1;
   struct point pt2;
};
```



Sada možemo deklarisati promenljivu screen

```
struct rect screen;
```

pa bi onda koordinata x prve tačke pravougaonika bila

```
screen.pt1.x;
```

Primer 1:

Napisati program koji izracunava obim i površinu trogula i kvadrata u koordinatnoj ravni.

```
#include <stdio.h>
#include <math.h>
struct point {
   int x;
   int y;  };
float segment_length(struct point A, struct point B) {
   int dx = A.x - B.x;
   int dy = A.y - B.y;
   return sqrt(dx*dx + dy*dy);
}
float Heron(struct point A, struct point B, struct point C) {
   float a = segment_length(B, C);
   float b = segment_length(A, C);
   float c = segment_length(A, B);
   float s = (a+b+c)/2;
   return sqrt(s*(s-a)*(s-b)*(s-c));
}
```

```
float circumference(struct point polygon[], int num) {
   int i;
   float o = 0.0;
   for (i = 0; i<num-1; i++)
      o += segment_length(polygon[i], polygon[i+1]);
   o += segment_length(polygon[num-1], polygon[0]);
   return o;
}

float area(struct point polygon[], int num) {
   float a = 0.0;
   int i;
   for (i = 1; i < num -1; i++)
      a += Heron(polygon[0], polygon[i], polygon[i+1]);
   return a;
}
```

```
main() {
   struct point a, b = {1, 2}, triangle[3];
   struct point square[4] = {{0, 0}, {0, 1}, {1, 1}, {1, 0}};
   a.x = 0; a.y = 0;
   triangle[0].x = 0;   triangle[0].y = 0;
   triangle[1].x = 0;   triangle[1].y = 1;
   triangle[2].x = 1;   triangle[2].y = 0;

   printf("sizeof(struct point) = %ld\n", sizeof(struct point));

   printf("x koordinata tacke a je %d\n", a.x);
   printf("y koordinata tacke a je %d\n", a.y);
   printf("x koordinata tacke b je %d\n", b.x);
   printf("y koordinata tacke b je %d\n", b.y);

   printf("Obim trougla je %f\n", circumference(triangle, 3));
   printf("Obim kvadrata je %f\n",  circumference(square, 4));
   printf("Pov. trougla: %f\n", Heron(triangle[0], triangle[1],
triangle[2]));
   printf("Pov. kvadrata: %f\n", area(square,
sizeof(square)/sizeof(struct point)));
}
```

Primer 3:

Napisati program kojim se učitavaju podaci za dve osobe i ispisuju podaci o starijoj. Podaci koji se znaju o osobi su: ime, adresa i starost (adresa može imati više reči).