

Programarea pe obiecte

1) Notiunile de obiect si metoda

Ideea de obiect are la baza o notiune mai simpla, deja cunoscuta, si anume aceea de inregistrare (*record*). Asa cum stim, o inregistrare este o colectie de date, de tipuri diverse, pe care programatorul le reuneste sub un singur nume conform unor anumite scopuri. Spre deosebire de inregistrari, obiectele pot contine, pe langa date, functii si proceduri.

Una dintre cele mai simple definitii de obiect este urmatoarea:

```
type complex=object  
    re,im:real;  
    function modul:real;  
end;  
function complex.modul:real;  
    begin  
    modul:=sqrt(sqr(re)+sqr(im));  
    end;
```

Notiunea de obiect reprezinta un concept superior celui de inregistrare prin faptul ca poate contine atat date (variabile) cat si metode.

Metodele sunt functii sau proceduri care fac parte din definitia obiectului si care sunt concepute de programator in asa fel incat sa reprezinte cat mai bine o anumita notiune.

Metodele au acces la membrii obiectului in mod implicit.

Parametrii formali ai metodelor nu pot coincide cu variabilele cuprinse in definitia obiectului.

Functiile membre ale unui obiect prelucreaza datele membre ale obiectului care apeleaza metoda.

2) Metoda constructor

Aceasta metoda este o metoda in sens obisnuit, care poate fi inclusa in definitia obiectului. La fel ca si alte metode, ea poate avea sau nu parametrii.

- Definitia unui obiect in Pascal reprezinta un tip de date creat de utilizator (programator) conform propriului scop.
- Definitia unui obiect este cuprinsa intre cuvintele rezervate ale limbajului `object` si `end`.
- Un obiect poate contine date (variabile) si metode (functii sau proceduri).

- Metodele sunt "anuntate" in definitia tipului de obiect si sunt descrise explicit in exteriorul acesteia.
- Metodele au acces implicit la restul membrilor obiectului.
- Metodele (functii sau proceduri) pot avea parametrii formali dar acestia trebuie sa difere de datele membre ale obiectului.

3) Incapsulare

Proprietatea obiectelor, prin care intr-o singura definitie sunt incluse date si metode, poarta numele de incapsulare.

Un unit este descris printr-un text sursa obisnuit in limbajul Pasca, intr-un fisier cu extensia pas si avand numele ales exact la fel cu cel al unit-ului pe care vrem sa il cream.

Iata care sunt regulile care trebuiesc respectate:

- 1) Prima linie contine numele pe care vrem sa-l dam unit-ului .
- 2) Inainte de prima definitie trebuie plasat cuvantul interface.
- 3) Dupa interface pot fi plasate directive uses cu sensul cunoscut, de utilizare a unor anumite functii, definitii sau proceduri.
- 4) Urmeaza definitiile tipurilor de obiecte.
- 5) Sectiunea care contine definitia sau definitiile de obiecte, se incheie prin cuvantul implementation.
- 6) Urmeaza descrierea propriu-zisa a metodelor.
- 7) In final, intre begin si end pot fi plasate instructiuni prin care, de exemplu, pot fi atribuite valori unor variabile anterior definite in unit-ul respectiv.
- 8) Intr-un unit pot fi incluse si alte definitii de functii sau proceduri, fara legatura cu tipul obiect, respectand conditia de anuntare a antentului in sectiunea interface si descrierea completa in sectiunea implementation.

4) Mostenire

Procesul de mostenire isi justifica denumirea chiar prin sensul original al cuvantului. Vom vorbi despre stramosi si urmasi. Urmasii preiau caracteristici ale stramosilor dar le pot modifica sau pot adauga achizitii noi.

Mostenirea reflecta preluarea tuturor datelor si metodelor unui stramos la care se pot aduga alte date si metode noi, rezultand un urmas. Metodele mostenite pot fi modificate de urmas.

5) Polimorfism

5.1 Rescrierea metodelor

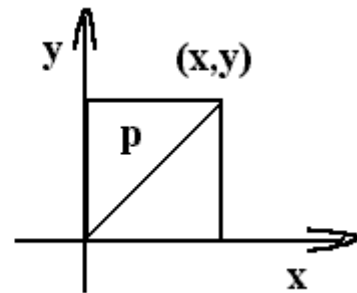
Urmasii pot modifica metodele mostenite. Simpla reanuntare a metodei in definitia urmasului conduce la inlocuirea metodei mostenite cu una noua.

Obiectul de tip punct va contine pe langa cele doua date ,urmatoarele metode : constructor,distanta pana la origine si afisare.

Metoda constructor init are doi parametri ale caror valori se transfera coordonatelor, cu alte cuvinte, prin initializare, specificam coordonatele punctului.

Metoda ro furnizeaza distanta pana la origine cu formula deja cunoscuta iar metoda de afisare afiseaza punctul in forma (x,y).

```
type punct=object  
  x,y:real;  
  constructor init(a,b:real);  
  function ro:real;  
  procedure afis;  
end;
```



Numim poliformism , acea prorpietate prin care o functie apelanta prin același nume, poate avea actiuni diferite de la stramos la urmas.

5.2 Concepte rafinate:inherited,virtual si self

Poliformismul prezentat in sectiunea anterioara are in limbajul Pascal doua rafinari extrem de interesante, pe care le vom prezenta separat.

In cele ce urmeaza vom ilustra o caracteristica a programarii pe obiecte in acest limbaj, si anume posibilitatea ca o functie mostenita dar modificata de catre urmas sa fie totusi accesibila.

Metodele virtuale sunt acele metode care se asociaza obiectului la momentul executiei.