

Introducere in programarea orientata pe obiecte (POO)

→ Oprian Adrian-George

Privire generala

Programarea Orientată Obiect (POO) este o metodă de proiectare și implementare în care programele sunt reprezentate sub forma unor colecții de obiecte, clase, care interacționează între ele prin intermediul mesajelor.



Principiile de baza ale POO

- Obiecte
- Clase
- Mostenire
- Ascunderea infomatiei
- Polimorfism
- Sabloane



Obiectele

- Sunt grupate in clase (obiecte cu aceleasi proprietati)
- Au attribute prin care le putem descrie starea
- Au actiuni (metode) prin care le putem modifica starea curenta
- Obiectele cu aceleasi proprietati sunt grupate in clase



Clasele

- Reprezinta un "tipar" dupa care sunt create obiectele
- Mentioneaza prorietaatile generale ale obiectelor
- Ascund informatia



Mostenirea

- Este un concept care confera putere unui limbaj orientat pe obiecte
- Permite extinderea unei clase fara a duplica codul existent



Ascunderea informatie

In momentul in care definim o clasa putem alege modul in care attributele unui obiect pot fi accesate din afara clasei. Accesul la informatia din clasa se face cu ajutorul a trei specificatori:

public, protected si private.

Specificatorul public permite accesul la attributele obiectului din orice parte a programului.

Specificatorul protected permite accesul doar din interiorul clasei si din clasa derivata.

Cu specificatorul private avem acces la attributele obiectului doar din interiorul clasei;



Polimorfismul

Polimorfismul este un concept prin care mai multe functii sau o metode pot avea acelasi nume, si functionalitate, diferenta la apel facandu-se prin tipul si numarul paramatrilor.

Un exemplu foarte util este o functie care calculeaza maximul a doua numere: `int max (int a,int b)/double max(double a, double b)`. Aici putem obseva ca functia max poate fi apelata atat pentru int cat si pentru double.



Sabloane

In spiritul programarii orientate pe obiect, sabloanele vin ca o completare si permit reutilizarea de cod, atat functii cat si clase. Scopul acestora este de a scrie cod cat mai general astfel incat sa eliminam erorile care ar putea aparea prin duplicarea de cod, adica vom scrie cod doar o data si acesta este corect il vom putea reutiliza fara a aduce modificari care tin de functionarea lui.

