

Incapsularea folosind clase in limbajele c++, python si java

C++

Incapsulare = mecanismul oferit de limbaj pentru a lega functii de date. Accesorii spre campuri si spre metode se face ca la structuri (in C cu punct). Variabilele de tipul clasa la nastere genereaza apelul de constructor, iar la moarte, destructorul.

In POO o **clasa este un tip de date**, iar **obiectele sunt instante** ale acestui tip de date.

Incapsularea in C++ nu reprezinta altceva decat proprietatea claselor de obiecte de a grupa sub aceeasi structura datele si metodele aplicabile asupra datelor. In sensul programarii orientata pe obiecte, incapsularea defineste, de asemenea, modalitatea in care diversele obiecte si restul programului se pot referi la datele specifice obiectelor. Casele din C++ permit separarea datelor in date private si date publice. Programele utilizatorilor pot accesa si utiliza datele unui obiect, declarate private, numai prin utilizarea unor metode definite public. Separarea datelor in sectiuni de date publice si private determina protejarea datelor fata de utilizarea lor eronata in programe

JAVA

Pe scurt, încapsularea este procesul prin care “înfășurăm” câmpurile unei clase prin niște metode cu scopul de a avea mai mult control asupra valorilor acestora. În mod normal, acest procedeu este folosit pentru câmpurile declarate *private*.

```
public class Caine{
    private String nume;
    private int lungimeCoadă;
    public String getNume(){
        return nume;
    }
    public void setNume(String nume){
        this.nume = nume;
    }
    public void getLungimeCoadă(){
        return lungimeCoadă
    }
}
```

```

public int setLungimeCoadă(int lungimeCoadă){
    this.lungimeCoadă = lungimeCoadă; }
}

```

Python

Incapsularea exprimă proprietatea de opacitate a obiectelor cu privire la structura lor internă și modul de implementare a metodelor. Fiecare element al sistemului poate executa acțiuni, își poate modifica starea și poate comunica cu alte elemente fără a dezvălui facilitățile detinute.

La nivelul clasei Persoana putem implementa incapsularea prin prefixarea numelui atributelor de instanță (nume, varsta) cu underscore. Limbajul Python protejează membrii declarați cu dublu underscore prin modificarea numelor acestora cu numele clasei.

```

class Persoana():
    """Clasa Persoana"""

    număr_persoane = 0;

    def __init__(self, nume, varsta):
        self.__nume = nume
        self.__varsta = varsta
        __class__.număr_persoane += 1

    def __str__(self):
        return "{0}, {1}, {2}".format(self.__nume, self.__varsta, __class__.număr_persoane)

    def se_prezintă(self):
        print("Bună! Numele meu este {} și am {} ani.".format(self.get_nume(), self.__varsta))

    def set_nume(self, nume):
        self.__nume = nume

    def get_nume(self):
        return self.__nume

```

EX 2

```
class student
{
    public:

        char nume[40];
        char facultate[40];
        char specializare[40];
        int varsta;
        double nota;

        void inregistrare_student(char *numestudent,char
*facultatestudent,char *specializarestudent,int *vs,double *nt);

        void inregistrare_student ();
};
```