

Programare Orientată Obiect

Laborator 3



Despre ce vorbim azi



- Pointer-ul `this`
- Funcții de acces - `get` și `set`

Pointerul this



- Metodele pot accesa datele membre fără să le primească drept parametri așa cum am văzut în laboratorul trecut
- Dacă privim lucrurile pur teoretic acest lucru este normal pentru că și datele și metodele aparțin aceleiași clase
- În practică însă la apelul metodelor se transmite un parametru în plus față de cei de apel
- Acesta conține adresa obiectului curent și este numit **this**
- **this** este cuvânt cheie și poate fi folosit de utilizator pentru aflarea adresei obiectului curent
- Chiar dacă acest lucru nu este vizibil direct, **this** este transmis ca prim parametru în orice apel de funcție membră

Exemple



```
• void student::afiseaza_adresa()
{
    cout<<"Adresa obiectului este: "<<this;
}
• char* student::returneaza_nume()
{
    return this->nume;
}
```

Funcțiile de acces (1)



- Teoria spune că cel mai bine e ca majoritatea atributelor să fie în domeniul de vizibilitate private (în fond acesta este și cel implicit)
- Tot teoria ne spune însă că putem asigura totuși acces la aceste date, dar într-un mod controlat
- Acest mod controlat se transpune în practică prin funcții ce nu au decât rolul de a returna sau modifica valoarea unui astfel de câmp
- Aceste funcții pentru a se distinge de celelalte pot fi prefixate de un șir de caractere sugestiv, recomandarea fiind `get_`*numeatribut* și `set_`*numeatribut*

Funcțiile de acces (2)



- Dacă ne amintim de laboratorul trecut funcția *returneaza_nume* nu făcea altceva decât să returneze numele studentului curent, adică era o funcție de tip *get_...*
- Asemenea ei putem declara o funcție *modifica_nume* ce va modifica numele studentului curent, fiind astfel o funcție de tip *set_...*
- Totuși ce rost mai are să declarăm attribute private dacă oricum le putem citi și modifica prin funcții membre?! Răspunsul îl vedem în continuare...

Avantajele funcțiilor de acces



- Putem seta attribute ca read only prin definirea metodei get, fără definirea metodei set; în acest fel valoarea câmpului va putea fi doar citită nu și modificată
- Prin funcții de acces putem returna o valoare modificată, nu pe cea reală (să ne gândim la un IBAN ca membru al clasei student, dacă cineva vrea să știe IBAN-ul am avea o breșă în securitate dacă l-am afișa integral, așa că, la fel cum se preferă în practică, chiar dacă IBAN-ul este definit complet în obiect, în funcția de tip get putem înlocui ultimele 5 caractere cu x-uri)
- Ne putem alege ce valoare să afișăm în funcție de tipul clasei care dorește informația; de exemplu dacă un angajat oarecare al facultății, care nu este profesor, dorește să știe media unui student, îi putem afișa caracterul confidențial al acesteia (un mesaj de eroare) și îi putem returna media întregului an din care face parte studentul

Clase pentru jocul de X și 0



- Exemplu practic

Temă facultativă



- Realizarea jocului de X și 0 propriu-zis pe baza claselor create