

Toate cerintele de teorie pentru POO - va rog sa completati/corectati ce stiti

Examen POO 2009 grupa 131.pdf - raspunsuri la subiecte teorie

II. Descrieti pe scurt cum este implementata mostenirea virtuala si scopul in care este folosita.

- mostenirea virtuala are urmatoare structura `class Derivata : virtual Baza { }`
- mostenirea virtuala este folosita in special pentru a evita ambiguitatea atunci cand avem doua clase diferite care deriva dintr-o baza si a patra care deriva din cele doua; in acel moment compilatorul nu stie care constructor sa aleaga pentru clasa de baza .

V. Spuneti ce este obiectul implicit al unei metode si descrieti pe scurt proprietatile pe care le cunoasteti despre acesta.

- obiectul implicit al unei metode este obiectul (cu care lucreaza) la care are acces metoda respectiva **este corect asa ?**
- proprietati :

sau

- ↑ -obiectul care apeleaza functia
- modificari produse in functie se reflecta asupra obiectului(e transmis prin referinta)
- eventualele apeluri/modificari se pot face apeland direct atributele/metodele sau prin pointerul `this`
- nu se mai apeleaza constructor/destructor pentru acesta

VIII. Descrieti pe scurt functiile virtuale si scopul in care sunt folosite.

- functiile virtuale sunt functiile care au cuvantul cheie 'virtual' in declararea lor; functiile virtuale se declara in clasa de baza si se definesc in clasa/clasele derivate
- functiile virtuale sunt descrieri generice pentru interfata comuna claselor de baza si derivate
- functiile virtuale se comporta ca functiile redefinite

XI. Descrieti pe scurt functiile statice si scopul in care sunt folosite.

Functiile statice pot fi aplicate tuturor obiectelor din acea clasa. Functiile statice pot adresa doar date statice din acea clasa, cele ne-statice sunt inaccesibile, de asemenea functiile statice nu pot apela functii nestatice din clasa respectiva. Functiile statice care sunt declarate in sectiunea public pot fi apelate fara specificarea unui obiect al clasei. Ele nu pot avea pointer de tip "this". Acestea au aplicatii limitate, dar o buna utilizare a lor este aceea ca ele pot "preinitializa" datele particulare de tip static, inainte de crearea efectiva a unui obiect.

XIV. Descrieti pe scurt diferenta dintre transferul parametrilor unei functii prin valoare si prin referinta constanta.

XVII. Descrieti pe scurt cum se comporta destructorii la mostenire.

- la mostenire este apelat intai destructorul clasei derivate apoi destructorul clasei de baza in mod invers constructorului

Examen POO 2009 grupa 151.pdf

2. Crearea dinamica de obiecte

- obiectele se pot crea dinamic folosind instructiunea 'new' pentru

alocarea memoriei in momentul executiei in heap: `class A { } ; main () { A variabila=new A(); }`

5. Proprietatile campurilor statice

- campurile statice nu depind de o instanta a clasei respective, se pot accesa direct
- campurile statice sunt vizibile in toate instantele clasei unde acestea au fost declarate
- campurile statice se initializeaza in afara clasei unde sunt declarate
- campurile statice ocupa o singura zona de memorie indiferent de numarul de obiecte declarate
- campurile statice sunt independente de obiectele din clasa si sunt create la lansarea programului

8. Proprietatile destructorului

- destructorul este o functie interna si obligatorie unei clase
- destructorul poate fi definit de programator in cazul in care nu este definit , compilatorul asigura un destructor implicit
- destructorul implicit distruge/sterge/elibereaza memorie unei variabile de tip cunoscut (int, float, double)
- destructorul se apeleaza in mod automat la iesirea din blocul de instructiuni al obiectului din clasa respectiva
- intr-o clasa putem avea un singur destructor
- ordinea de executie a destructorilor este inversa ordinii de executie a constructorilor (destructorii se executa de la clasele derivate spre clasa de baza)
- nu pot returna o valoare
- nu trebuie sa posede lista de argumente
- pot fi virtuali

11. Descrieti pe scurt diferenta dintre parametrii transmisi prin pointeri si cei prin referinta.

-

14. Ce reprezinta o functie virtuala si in ce conditii o functie virtuala defineste o clasa abstracta.

-

17. RTTI - (mecanismul de control al tipului in timpul executiei) = (run time type identification la pointeri)

- <https://www.facebook.com/groups/228669077176404/405786119464698/>

Examen P00 2009 grupa 153.pdf

V Descrieti pe scurt diferenta dintre o clasa si un obiect

- un obiect este o instanta a unei clase
- o clasa reprezinta un tip de date abstract care contine metode si tipuri de date

XIV. Descrieti pe scurt funciile sablonsi dati exemplu de 3 situatii in care un apel de functie NU genereaza o versiune a functiei dintr-un sablon disponibil pentru functia respectiva.

XVII. -Descrieti pe scurt diferenta dintre polimorfismul de compilare si cel de executie.

Restanta 2009.pdf

II. Descrieti pe scurt in ce consta mecanismul de incapsulare

-

XIV . Descrieti pe scurt mecanismul de tratarea a exceptiilor

-

XVII . Descrieti pe scurt diferenta dintre un pointer si o referinta

-

PPOO_Modeldesubiect_iunie2005.pdf

II. Descrieti pe scurt diferența dintre funcțiile care returnează valoare și cele care returnează referință.

-

V. Descrieti pe scurt cum se pot defini funcții de conversie între tipuri (clase).

-

VIII. Spuneți care este diferența dintre clasa generică (template) și clasa abstractă și in ce situații se folosește fiecare dintre ele.

Clase abstracte-Declararea unei clase abstracte.Se face folosind cuvântul rezervat *abstract* în declaratia clasei: `abstract class ClasaAbstracta { . . . }`

- Daca vom încerca sa instantiem un obiect al clasei ClasaAbstracta vom obtine o eroare la compilarea programului

Metode abstracte-Spre deosebire de clasele obisnuite care trebuie sa furnizeze implementari pentru toate metodele declarate o clasa abstracta poate contine metode fara nici o implementare. Metodele fara nici o implementare se numesc metode abstracte si pot aparea doar în clase abstracte. In fata unei metode abstracte trebuie sa apara cuvântul cheie *abstract*, altfel va fi furnizata o eroare de compilare.

Diferenta dintre o clasa abstracta si si o clasa generica consta in urmatoarele proprietati : o calsa abstracta nu se poate instantia pe cand o clasa template da

XIV. Spuneți pe scurt prin ce se caracterizează o metodă statică a unei clase.

XVII. Spuneți care este diferența dintre incluziunea de clase și moștenirea de clase și când se folosește fiecare metodă.

- incluziune (vb “a avea”), mostenire(vb “a fi”)

PPOO_Modeldesubiect_ianuarie2007.pdf

VIII. Spuneți ce reprezintă o funcție prietenă (friend) a unei clase.

-functia friend este o functie externa care are acces la datele private si protejate ale clasei unde este declarata aceasta

-

XI. Spuneți dacă o variabilă constantă poate fi transmisă ca parametru al

unei funcții și dacă da, în ce situații. Justificați.

-deci o variabila const poate fi trimisa ca parametru pt o functie dar cu anumite restrictii. o data ce faci apel prin valoare cu o const parametrul formal nu se schimba in functie. de asemenea cand e constant la la intoarcere , valoarea returnata nu se poate schimba. daca se transmite o adresa valoarea nu se poate schimba nici in acest caz.

exp: void f1(const int i) {i++;} <- NASPA RAU eroare

void f2(int ic) {const int&i=ic; i++;} <- iar eroare!!!

sper ca e ok. corectati daca e gresit.

XVII. Descrieți pe scurt ce reguli verifică supraîncărcarea operatorilor.

1. Nu putem defini operatori noi. Doar supraincarcam operatori deja existenti.
2.Nu putem modifica numarul operanzilor unui operator existent si nici sintaxa lor.

3. Nu putem modifica prioritatea si nici asociativitatea operatorilor existenti.

4. **Exista operatori care nu se pot supraincarca (exemplu: ".", ":", "?:", ".*".)**

5.Operatorii pot fi supraincarcati ca metode sau ca functii independente.

Nu toti operatorii care pot fi supraincarcati ca metode pot fi supraincarcati ca functii independente ! (exemplu: "=", "(", "[", ">")

PPOO_Modeldesubiect_iunie2006.pdf

II. Descrieți pe scurt constructorul de copiere și situațiile în care acesta este folosit..

Constructorul de copiere este un constructor cu un singur parametru de tipul clasei. El are o forma implicita care copiaza bit cu bit.

Utilizarea constructorului de copiere :

- 1)Pentru initializarea obiectelor din alte obiecte de acelasi tip
- 2)La transferul parametrilor unei functii prin valoare
- 3)La intoarcerea rezultatului unei functii prin valoare.

XI. Descrieți pe scurt cum puteți prelua o dată prin incluziune și a doua oară prin moștenire o clasă *numar* într-o clasă *lista* care descrie liste nevide de dimensiune variabilă de elemente de tip *numar*.

prima var, prin compunere, ca la liste, cand aveai nod, doar ca acum ai o clasa numar in loc de nod.

si mai e o var cand in numar ai informatiile dar nu si legatura, mostenesti in lista numar si ai acolo legatura spre urm lista adica lista *next,

PPOO_Modeldesubiect_iunie2008.pdf

II. Descrieti trei metode de proiectare diferite prin care elementele unei clase se pot regăsi în dublu exemplar, sub diverse forme, în definitia altei clase.

Nu stii sigur de a treia, dar doua sunt: prima prin mostenire multipla, nevirtuala(cand mostenesti o clasa si inca una care e derivata din prima clasa mostenita), a doua prin compunere, de ex poti sa ai doua obiecte din doua clase

diferite care mostenesc aceeași clasă de bază și a treia prin mostenire și compunere, a aceiași clase de bază.

XI. Enumerați 3 metode de implementare a polimorfismului de compilare.

Supraincarcare, Sabloane și Parametri cu valori implicite

Explicați `dynamic_cast`