

Programare Orientată Obiect = Proiecte 2013-2014 =

Formulare generală:

Realizarea unui proiect în care să fie implementată o ierarhie de clase într-o aplicație consolă. Vor putea fi create obiecte de tipurile specificate, printr-un mecanism de tip meniu care să permită specificarea tipului obiectului creat, cu parametrii specifici și, odată cu crearea, să se salveze conținutul lor într-un fișier. Suplimentar, în orice moment vor putea fi listate obiectele aflate în memorie.

Scop general:

Proiectul este conceput în scopul verificării cunoștințelor de programare dobândite și dezvoltării capacității individuale de a realiza o temă de complexitate medie. Proiectul se realizează individual.

Scop specific:

- înțelegerea conceptelor de programare orientată obiect în special aspectele de derivare, polimorfism și lucru cu stream-uri;
- implementare în limbaj C++ a unei ierarhii de clase, definirea și implementarea funcționalităților necesare; scrierea unor programe cu o interfață în linie de comandă cu un meniu de tip text (cu opțiuni 1, 2, 3 ...);
- deprinderea cu particularitățile lucrului în mediul de dezvoltare Visual C++ 6: lucrul cu proiecte, utilizare editor/compiler, depanarea aplicațiilor prin execuție pas cu pas, inspectarea variabilelor;
- deprinderea cu utilizarea MSDN pentru regăsirea informațiilor utile;

Documentație specifică:

- carte C/C++; recomandarea noastră H. Schildt, The complete reference C++, 3rd edition, ed. McGraw Hill, 1999, ISBN 0-07-882476-1;
- Microsoft MSDN 6, 2CD sau web site;
- lucrări de laborator la disciplinele PCLP 1 & 2 și POO, Fac. ETTI;

Cerințe: Proiectul se prezintă atât în format electronic, pe un CD-ROM sau memorie externă USB, eventual se poate veni cu laptop-ul propriu, cât și scris, într-o mapă conținând:

- tema de proiect;
- scurta descriere a conținutului proiectului și a fișierelor sursa;
- conținutul fișierelor sursa complet;
- scurtă descriere a programului de test;
- programele de test realizate și mesajele afișate la consolă.

Notarea se face pe baza RULĂRII/TESTĂRII CODULUI PROIECTAT, apreciind soluția tehnică aleasă, rezolvarea corectă și completă a problemei, forma prezentării scrise (claritate, corectitudine, argumentare), susținerea orală a proiectului și capacitatea de a răspunde corect la întrebări tehnice privitoare la proiect.

Termen de predare și susținere: LA LABORATOR, ÎN SĂPTAMÂNA a 14-a, ÎN CONFORMITATE CU ORARUL. PREZENTAREA ULTERIOARĂ SE PENALIZEAZĂ CU 2 PUNCTE!

1. Tema proiect nr. 1

Să se definească adecvat o clasă abstractă cu numele `Fruct`, din care să se deriveze public clasele `Mar` și `Para`.

Clasa `Fruct` va avea ca dată membru privată greutate (`double`), clasa `Mar` va avea data privată soi (`Șir`), iar clasa `Para` va avea dată membru privată grosime_coaja (`int`). Tipul de date `Șir` va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructor și o funcție care să permită scrierea unui mesaj de identificare a clasei într-un fișier (cu nume constant). Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

2. Tema proiect nr. 2

Să se definească adecvat o clasă abstractă cu numele `Triunghi`, din care să se deriveze public clasele `Isoscel` și `Echilateral`.

Clasa `Triunghi` va avea ca date membru private `Arie` (`int`) și `Perimetru` (`int`), clasa `Isoscel` va avea data privată obligatorie `unghi_bază` (`double`), iar clasa `Echilateral` va avea dată membru privată `raza_cerc_circumscribit` (`double`).

În mod adecvat, pentru clasa de bază (după caz) și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

3. Tema proiect nr. 3

Să se definească adecvat o clasă abstractă cu numele `Mobila`, din care să se deriveze public clasele `Biblioteca` și `Birou`.

Clasa `Mobila` va avea ca dată membru privată producător (`Șir`), clasa `Biblioteca` va avea data privată `nr_carti` (`int`), iar clasa `Birou` va avea dată membru privată `nr_sertare` (`int`). Tipul de date `Șir` va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

4. Tema proiect nr. 4

Să se definească adecvat o clasă abstractă cu numele Consumabil, din care să se deriveze public clasele Toner și Hartie.

Clasa Consumabil va avea ca dată membru privată producător (Șir), clasa Toner va avea data privată pret_cartus (int), iar clasa Hartie va avea date membru private pret_coala (int) și calitate (char). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

5. Tema proiect nr. 5

Să se definească adecvat o clasă abstractă cu numele Tanar, din care să se deriveze public clasele Student și Sportiv.

Clasa Tanar va avea ca date membru private scoli_absolvite (Șir), clasa Student va avea data privată nume_institutie (Șir), iar clasa Sportiv va avea dată membru privată nume_club (Șir). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

6. Tema proiect nr. 6

Să se definească adecvat o clasă abstractă cu numele Carte, din care să se deriveze public clasele Proza și Versuri.

Clasa Carte va avea ca date membru private nr_pagini (int) și autor (Șir), clasa Proza va avea data privată nr_capitole , iar clasa Versuri va avea dată membru privată tip_rima (enum). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

7. Tema proiect nr. 7

Să se definească adecvat o clasă abstractă cu numele Verificare, din care să se deriveze public clasele Grila și Proiect.

Clasa Verificare va avea ca dată membru privată grad_dificultate (enum), clasa Grila va avea date private nr_intrebări (int) și timp în minute (int), iar clasa Proiect va avea dată membru privată titlu_proiect (Șir). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

8. Tema proiect nr. 8

Să se definească adecvat o clasă abstractă cu numele Animal_companie, din care să se deriveze public clasele Pișica și Caine.

Clasa Animal_companie va avea ca dată membru privată booleana de_rasa, clasa Pișica va avea date membru private lungime_par (enum) și vârsta (int), iar clasa Caine va avea dată membru privată rasa (Șir). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

9. Tema proiect nr. 9

Să se definească adecvat o clasă abstractă cu numele Telefonie, din care să se deriveze public clasele TelefonieClasică și TelefonieIP.

Clasa Telefonie va avea ca dată membru privată furnizor (Șir), clasa TelefonieClasică va avea dată membru privată lățime_banda în Kb/s (int), iar clasa TelefonieIP va avea dată membru privată tip_transport (enum), de ex. TCP/IP, ADSL, HDSL, etc. Tipul de dată Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

10. Tema proiect nr. 10

Să se definească adecvat o clasă abstractă cu numele Functionar, din care să se deriveze public clasele Casier și Secretara.

Clasa Functionar va avea ca dată membru privată un vector cu 3 atribuții principale (Șir), clasa Casier va avea dată membru privată valoare_lei (int), iar clasa Secretara va avea dată membru privată ora_audienta (int). Tipul de dată Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi

mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

11. Tema proiect nr. 11

Să se definească adecvat o clasă abstractă cu numele Tablou, din care să se deriveze public clasele Vector și Matrice.

Clasa Tablou va avea ca dată membru privată numele sau (Șir), clasa Vector va avea dată membru privată nr_el (int), iar clasa Matrice va avea date membru private nr_lin și nr_col (int). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

12. Tema proiect nr. 12

Să se definească adecvat o clasă abstractă cu numele Carte, din care să se deriveze public clasele Roman și Nuvela.

Clasa Carte va avea ca date membru private editura (Șir) și nr_pagini (int), clasa Roman va avea dată membru privată tip_roman (enum care cuprinde: AVENTURI, POLITIST, etc), iar clasa Nuvela va avea dată membru privată bibliografie_obligatorie (bool). Tipul de data Șir va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

13. Tema proiect nr. 13

Să se definească adecvat o clasă abstractă cu numele `Mijloc_Transport`, din care să se deriveze public clasele `Tramvai` și `Mașina`.

Clasa `Mijloc_Transport` va avea ca dată membru privată `in_comun` (bool), clasa `Tramvai` va avea dată membru privată `nr_calatori`, iar clasa `Mașina` va avea dată membru privată `cai_putere` (int).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.

14. Tema proiect nr. 14

Să se definească adecvat o clasă abstractă cu numele `Insecta`, din care să se deriveze public clasele `Future` și `Gandac`.

Clasa `Insecta` va avea ca dată membru privată `familia` (Șir), clasa `Future` va avea dată membru privată `tzara_provenienta` (Șir), iar clasa `Gandac` va avea dată membru privată `de_bucatarie` (bool).

Tipul de dată `Șir` va fi un tip utilizator (clasă separată).

În mod adecvat, pentru clasa de bază și clasele derivate se vor defini constructori, constructorul de copiere, supraincercarea operatorilor de inserție și extracție, destructori. Să se construiască cele două clase astfel încât să poată fi exploatat polimorfismul.

Să se scrie un program care să evalueze comportamentul polimorfic, definind corespunzător un tablou cu elemente de tip adecvat, creat în funcție de opțiunea utilizatorului.

Să se proiecteze și apeleze adecvat o funcție care are ca argument un obiect din ierarhia de clase și, apelând la funcția de identificare, afișează un mesaj privitor la clasa careia îi aparține. Se va folosi mecanismul polimorfic astfel încât funcția să poată fi apelată pentru obiecte aparținând ambelor clase.