

Lista temelor, cu toate cerintele pentru a treia lucrare practica.

- Cerinte comune tuturor temelor:

1. tratarea exceptiilor
2. utilizarea conceptelor de Dynamic_Cast și Static_Cast
3. utilizarea sabloanelor
4. citirea informațiilor complete a n obiecte, memorarea și afisarea acestora

Obs. Se pot considera și probleme din fișierul Tema2.docx, adaptate la cerintele de mai sus.

- cerinte generale aplicate fiecărei teme din acest fisier:
 - să se identifice și să se implementeze ierarhia de clase;
 - clasele sa conțină constructori, destructori, =, >>, <<;
 - clasa de baza sa conțină funcție virtuala de afisare, rescrisa în clasele derivate;
 - clasele derivate trebuie sa contina cel puțin constructori parametrizati și de copiere prin care sa se evidentieze transmiterea parametrilor către constructorul din clasa de baza.

Alte probleme:

Tema 1. Firma X are un domeniu de business unde este necesar să se urmărească modul în care clientii plătesc (numerar, cec sau card de credit). Indiferent de modul de plata, firma X știe în ce data s-a efectuat plata și ce suma a fost primita. Dacă se plătește cu cardul, atunci se cunoaște și numărul cardului de credit. Pentru cash, nu e necesara identificarea clientului care a făcut plata.

Cerinta suplimentara:

- Sa se construiasca clasa template **Gestiune** care sa conțină numărul total de plati (incrementat automat la adaugarea unei noi chitante) și un vector de pointeri la obiecte de tip Plata, alocat dinamic. Sa se supraincarce operatorul += pentru inserarea unei plati în lista, indiferent de tipul acesteia.
- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa stocheze numarul de clienti, impreuna cu numele acestora.

Tema 2. La ora de Biologie, copiii din ciclul gimnazial învața ca regnul animal se împarte în 2 grupuri: nevertebrate și vertebrate. La rândul lor, vertebratele se împart în pesti, păsări, mamifere și reptile.

Cerinta suplimentara:

- Să se adauge toate campurile relevante pentru modelarea acestei probleme.

- Sa se construiasca clasa template **AtlasZoologic** care sa contină un număr de animale (incrementat automat la adaugarea unei noi file) și un vector de pointeri la obiecte de tip Regn, alocat dinamic. Sa se supraincarce operatorul += pentru inserarea unei fise de observatie a unui animal în lista, indiferent de tipul acestuia.
- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa contină și sa afiseze doar numarul de nevertebrate.

Tema 3. La facultatea Y studenții intra în sesiune. In regulament e prevăzut ca ei sa aibă un anumit numar de examene. Fiecare examen are un numar, incrementat automat, denumirea materiei data și nota la scris. Partialul conține și nota la oral, iar examenul final conține extra-puncte pe un proiect. Dacă partialul nu e luat, atunci se reface la examenul final, altfel, se păstrează nota. Cei care vor să-si mareasca nota, pe loc, mai dau un quiz, continant un numar de itemi de tip grila.

Cerinta suplimentara:

- Sa se construiasca clasa template **CatalogIndividual** care sa contină informații despre examenele date de un student. Clasa conține nr matricol al studentului (incrementat automat la adaugarea unei noi file), nr_examene și un vector de pointeri la obiecte de tip Examen, alocat dinamic. Sa se supraincarce operatorul += pentru inserarea unei fise de observatie a unui examen în lista, indiferent de tipul acestuia.
- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa contină și sa afiseze doar media la examenele finale.

Tema 4. Dintr-un parc auto se poate cumpăra o gama variata de automobile din urmatoarele clase: MINI (mașini de oraș de mic litraj, de obicei sub 4m lungime), MICA (mașini de oraș, cu spațiu interior mai mare decât MINI și lungime între 3.85 și 4.1), COMPACTA (mașini ușor de folosit atât de oraș cât și la drum lung, de dimensiune 4.2 – 4.5m; modelele vin sub forma de hatchback, combi sau sedan) și MONOVOLUME (automobile sub forma de van ce pot transporta 5-7 persoane). Monovolumele pot și achizitionate atât noi cât și second hand. La cele achizitionate sh se percepe un discount proportional cu numărul de ani vechime și, în lunile de vara, beneficiaza de zile promotionale cu reducere fixa de 10% din preț.

Cerinta suplimentara:

- Să se adauge campurile și metodele necesare pentru implementarea corecta;
- Să se construiasca clasa template **Vanzare** care sa contină nr total de mașini în stoc (decrementat automat la vanzarea unei masini), nr de mașini vândute (incrementat automat) și doi vectori de pointeri la obiecte de tip masina, alocati dinamic, prin care să se gestioneze automobilele din stoc și cele vandute. Sa se supraincarce operatorul -= care sa reactualizeze datele din clasa la vanzarea unei mașini, indiferent de tipul acesteia.

- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa conțină și sa afiseze gestioneze doar MONOVOLUMELE.

Tema 5. În luna mai se organizeaza târgul mașinilor sport, astfel ca pasionatii se pot delecta cu modele din clasa COUPÉ, COUPÉ cu 4 uși, HOT-HATCH (modele hatchback de clasa mica și compacta ci motoare performanțe ce au la baza modele obișnuite), CABRIO(modele decapotabile cu acoperiș metalic sau din material textil) și SUPERSPORT(masini de viteza gen Audi R8, Bugatti Veyron, Lexus LF-A, etc.). O parte din masinile supersport poate să fie vânduta chiar în cadrul expozitiei, dacă se tranzactia se face cu plata pe loc.

Cerinta suplimentara:

- Să se adauge campurile și metodele necesare pentru implementarea corecta;

- Să se construiască clasa template **Expozitie** care sa conțină nr total de mașini expuse (incrementat automat) și un vector de pointeri la obiecte de tip mașina, alocat dinamic.

- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa conțină nr total de mașini supersport expuse (decrementat automat la vanzarea unei masini), nr de mașini vândute (incrementat automat) și doi vectori de pointeri la obiecte de tip mașina supersport, alocati dinamic, prin care să se gestioneze automobilele din stoc și cele vandute. Sa se supraincarce operatorul -= care sa reactualizeze datele din clasa la vanzarea unei mașini.

Tema 6. Se doreste implementarea unei aplicatii OOP care sa permita gestionarea activitatii unor farmacii apartinand proprietarului X. Pentru fiecare farmacie se cunoaște cel puțin denumirea, numărul de angajați și profiturile pe fiecare luna. Farmaciile pot fi și online. În acest caz se cunoaște doar adresa web, numărul de vizitatori și discountul utilizat.

Cerinta suplimentara:

- Sa se construiasca clasa template **GestionareFarmacii** care sa conțină informații despre lantul de farmacii X. Clasa conține indexul farmaciei (incrementat automat la adaugarea unei noi file), id-ul lanțului (constant) și un vector de pointeri la obiecte de tip Farmacie, alocat dinamic. Sa se supraincarce operatorul += pentru inserarea unei noi locatii în lista, indiferent de tipul acesteia.

- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa conțină și sa afiseze doar numărul total de vizitatori ai farmaciilor online.

Tema 7. La realizarea unui film(nume,tip,durata), participa o mulțime de persoane: un regizor, actori, personal tehnic etc. Dacă filmul este bun, atunci el ruleaza într-un numar de cinematografe și realizeaza incasari substantiale pe fiecare cinematogram în parte. Fiecare personal (cnp, nume, nume_film) are prevăzut prin contract un procent din incasari, drept urmare primește, pentru fiecare film la care participa, o anumita suma.

Actorii, dacă sunt personaje principale, primesc un bonus de 10% din incasari, iar regizorul are prevăzut și o suma fixa, indiferent de calitatea filmului produs.

Cerinta suplimentara:

- Sa se construiasca clasa template **FirmaDistributie** care sa conțină informații despre filmele realizate într-un an. Clasa conțină numărul total de persoane implicat incrementat automat), numărul de actori (incrementat automat) și un vector de persoane alocat dinamic.
- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa conțină numărul actorilor care au fost distribuiti în roluri principale.

Tema 8 - Se doreste implementarea unei aplicatii care sa permita gestionarea conturilor deschise la banca X. Fiecare cont bancar are obligatoriu un detinator, o data a deschiderii lui si un sold. In cazul conturilor de economii, trebuie retinuta si rata dobanzii (care poate fi pe 3 luni, pe 6 luni sau la un an), precum si un istoric al reactualizarii soldurilor de la deschidere si pana in prezent. In cazul in care detinatorul opteaza pentru un cont curent, el beneficiaza de un numar de tranzactii gratuite si altele contra cost (de exemplu orice depunere este gratuita, dar retragerea poate sa coste daca s-a depasit numarul de tranzactii gratuite, sau e facuta de la bancomatele altor banci; sau orice cumparatura online are un cost, etc.). Simulati cat mai corect activitatea bancii X.

Cerinta suplimentara:

- Sa se construiasca clasa template **GestionareConturi** care sa conțină informații despre banca X. Clasa conține indexul unui cont (incrementat automat la adaugarea unui nou prin supraincarcarea operatorului +=) și un vector de pointeri la obiecte de tip ContBancar, alocat dinamic.
- Sa se construiasca o specializare a acestei clase pentru **un tip de data la alegere** care sa trateze exceptiile ce pot aparea de la utilizarea valorilor negative.

Tema 9 - Pizzeria X testeaza un nou soft, dezvoltat in maniera OOP, pentru gestionarea activitatii sale. Codul propriu-zis contine o clasa abstracta care contine doar metoda virtuala pura de calcul al pretului unui produs, iar din aceasta clasa de deriveaza clasa Pizza. In realizarea oricarui produs intra un anumit numar de ingrediente pentru care se cunosc denumirile, cantitatile si pretul unitar. Pretul unui produs finit este data de pretul ingredientelor plus manopera (o suma constanta fixata de producator). Daca pizza este comandata OnLine, la pret se mai adauga si 5% din pret la fiecare 10 km parcursi de angajatul care livreaza la domiciliu.

Cerinta suplimentara:

- Sa se construiasca clasa template **Meniu** care sa gestioneze tipurie de pizza comercializate. Clasa trebuie sa contina indexul unui produs

(incrementat automat la vanzarea unui produs prin supraincarea operatorului +=) și un vector de pointeri la obiecte de tip pizza, alocat dinamic. Tipurile de pizza vegetariana trebuie tratate într-un document contabil separat din care să rezulte valoarea totală încasată din vânzarea acestora.

- Să se construiască o specializare a acestei clase pentru **un tip de date la alegere** care să trateze excepțiile ce pot apărea de la utilizarea valorilor negative.

Tema 10 - Se dorește implementarea unei aplicații care să permită gestionarea clienților și a proprietăților imobiliare în cadrul unei agenții imobiliare nou înființate. Pentru fiecare locuință se cunoaște numele clientului care o închiriază, suprafața utilă și discount-ul (0-10%). La apartamente se cunoaște etajul, iar la case se știe câți metri pătrați are curtea, câte etaje are casa și care este suprafața utilă pe fiecare etaj în parte.

Evident, calculul chiriei se face diferit. Dacă la apartamente se consideră doar formula dată de prețul de închiriere pe metru pătrat * suprafața utilă, având grijă să se aplice discount unde este cazul, la case, se adaugă, indiferent de discount, prețul pe metru pătrat de curte * suprafața acesteia.

Cerință suplimentară:

- Să se construiască clasa template **Gestiune**, conținând un vector de pointeri la obiecte de tip **Locuință** alocat dinamic, unde indexul fiecărei locuințe este incrementat automat la adăugarea uneia noi, prin supraincarea operatorului +=. Clasa Gestiune trebuie să stocheze numărul de apartamente sau case, fiecare cu chiria lunară aferentă.
- Să se construiască o specializare a acestei clase pentru **un tip de date la alegere** care să stocheze numărul de clienți, împreună cu numele acestora.