

Bilet 1

Sa se implementeze tipul abstact *DataCalendaristica* care sa cuprinda:

- Constructor de initializare
- Functie statica pentru validare a unei date calendaristice
- Functie pentru citire si pentru afisarea unei date calendaristice cu un anumit format(zz/ll/aaaa sau ll/zz/aaaa sau aaaa/ll/zz)

Bilet 2

Sa se realizeze o clasa *DataCalendaristica* care sa contina o functie membru statica care primeste un obiect *DataCalendaristica* si intoarce un obiect de acelasi tip dar care contine ziua urmatoare.

Bilet 3

Sa se implementeze tipul abstact *Triunghi*. Clasa va contine :

- Constructor de initializare
- O functie membru statica care verifica daca lungimile laturilor pot forma un triunghi.
- Functii pentru calculul ariei si perimetrului
- O functie care verifica daca triunghiul este sau nu dreptunghic.

Bilet 4

Sa se implementeze tipul abstact *Coadă* ale carei elemente sunt numele unor persoane. Clasa va trebui sa contina:

- Constructor de initializare.
- Constructor de copiere.
- Destructor.
- Metode pentru adaugarea,eliminarea unei personae,pentru a testa daca coada este vida sau nu.

Bilet 5

Sa se implementeze tipul abstract *Polinom*. Clasa va contine:

- Constructor de copiere
- Destructor
- Metode pentru citire si afisare.

Sa se realizeze o functie friend care sa realizeze suma a doua polonoame.

Bilet 6

Sa se implementeze tipul abstract *Matrice* care va retine matricea sub forma unui tablou unidimensional. Clasa va contine:

- Constructor de initializare
- Constructor de copiere
- Destructor
- Metode pentru citire si afisare
- Metoda pentru calculul normei.

Bilet 7

Se considera o sala de cinematograful in care scaunele sunt asezate pe linii si coloane sub forma unei matrici. Un loc in sala este identificat prin rand si numarul de ordine al scaunului in cadrul randului. Sa se implementeze o clasa care sa retina diagrama locurilor pentru bilete vandute.

Indicatie: Clasa va avea metode pentru :

- Vanzare a unui bilet (care va furniza un loc in sala).
- Functie pentru anularea unei rezervari
- Afisarea ocuparii salii

Bilet 8

Sa se implementeze o clasa *Cursa*(numar_cursa, sursa, destinatie, ora_plecari). Si o clasa care sa tina evidenta curselor dintr-un aeroport care va cuprinde metode pentru:

- Listarea tuturor curselor cuprinse intre anumite ore
- Ordonare a zborurilor dupa ora plecarii si dupa destinatie

Bilet 9

Sa se implementeze clasa *Produs* (denumire,pret) si clasa *Magazin* care retine produsele dintr-un magazin. Clasa *Magazin* va contine metode pentru listarea ofertei de produse in care se vor afisa preturile in LEI si in USD.

Indicatie: Se va folosi o data membru statica care va retine paritatea leu/usd iar preturile se vor retine doar in dolari.

Bilet 10

Sa se implementeze tipul abstract *Complex* care sa contina:

- Constructor cu parametri impliciti
- O metoda friend care sa calculeze z^n folosind formula lui *Moivre*

Bilet 11

Sa se implementeze clasa numar rational de forma a/b. Programul va contine:

- Constructor cu un parametru implicit (utilizat pentru numerele intregi).
- O functie pentu simplificare.

O functie friend pentru a realiza suma a doua numere rationale

Bilet 12

Sa se implementeze clasa *echipa de fotbal* care retine obiecte de tip *Jucator*. Sa se afiseze componenta echipei si sa se implemnteze o functie pentru a realiza substitutii.

Bilet 13

Sa se implementeze o clasa in C++.

Bilet 14

Sa se implementeze clasa *Pacient* (nume, prenume, varsta, diagnostic). Clasa va contine:

- Constructor de initializare
- Constructor de copiere
- Destructuctor
- Metoda pentru citire
- Metoda pentru afisare

Sa se construiasca un tablou cu obiecte de tip *Pacient* si sa se ordoneze crescator dupa varsta.

Bilet 15

Sa se implementeze clasa *Produs*(denumire, pret, unitate de masura). Utilizati aceasta clasa pentru a implementa clasa *Oferta* care retine produsele oferite spre vanzare intr-un magazin. Sa se afiseze lista de preturi in lei si in \$.

Indicatie: Se va folosi o data membru statica care retine paritatea leu/\$

Bilet 16

Sa se implementeze clasa *String*. Clasa va contine:

- Constructor de copiere
- Destructuctor
- Functie pentru a determina lungimea sirului
- Functie care determina numarul de vocale din sir

Nu se vor utiliza functii din biblioteca *string.h*

Bilet 17

Sa se implementeze clasa *Imprimanta* care retine sub forma unei cozi obiecte de tip *Document*(nume,nr de pagini) care sunt trimise spre imprimare. Clasa va trebui sa aiba metode pentru:

- Adaugarea unui document in coada(la sfarsit)
- Listarea unui document (primul)
- Anularea imprimarii unui document cu un anumit nume.
- Afisarea continutului cozii.

Bilet 18

Sa se implementeze clasa *Matrice* care retine matrici *rare* (matrici cu foarte multe elemente 0). Se va utiliza o lista in care se retine: linia, coloana, elementul de pe acea pozitie. Clasa trebuie sa contina:

- Constructor de copiere
- Destructor
- Metoda pentru citire
- Metoda pentru afisare
- Modificarea unui element
- O functie friend care calculeaza suma a doua astfel de matrici.

Bilet 19

Sa se realizeze o clasa *Multime* ale carei elemente sunt numere intregi. Clasa va contine :

- Metoda pentru citirea unei multimi
- Metoda pentru afisarea unei multimi
- Metoda statica care valideaza o multime(testeaza daca toate elementele sunt distincte).
- Metoda friend care sa calculeze intersectia a doua multimi.

Bilet 20

Sa se realizeze o clasa *Email* care retine o scrisoare electronica. Clasa va contine urmatoarele campuri (*from* = adresa de email a persoanei care expediază scrisoarea, *to*= adresa de email careia ii este adresata scrisoarea, *subject* = scurta descriere a continutului, *body* = continutul scrisorii). Clasa va trebui sa valideze daca un sir de caractere este o adresa de email valida. (O adresa de email este valida daca este de forma: [nume@domeniu](#) ; unde *nume* poate fi orice combinatie de cifre si litere, caracterul underline (_) si punct si nu poate fi vid; *domeniu* este de forma *subsubdomeniu.subdomeniu.domeniu* (prin (sub(sub))domeniu se intelege orice combinatie de cifre si litere , _,-). Iar ultimul sir de caractere este de lungime 2 sau 3.

Bilet 21

Sa se implementeze clasa *Planeta* (nume, raza, distanta fata de soare, durata perioade de rotatie in jurul soarelui). Sa se implementeze apoi clasa *SistemSolar* care are un nume si lista planetelor.

Bilet 22

Sa se implementeze clasa *Numar* care retine un numar natural intreg in orice baza ca un vector al cifrelor. Sa se realizeze o functie friend care realizeaza suma a doua astfel de numere

Bilet 23

Sa se realizeze clasa *PartidPolitic* (nume partid, nume lider, orientare, numar membri,numar voturi). Sa se utilizeze a obiecte de acest tip pentru a afisa o statistica a unui sondaj de opinie cu privire la popularitatea partidelor la un moment dat.

Bilet 24

Sa se implementeze clasa *AgendaTelefonica* al carei continut sunt obiecte de tip *Abonat*(nume, adresa, numar de telefon). Sa se afiseze ordonata dupa nume si numar de telefon continutul agendei..

Bilet 25

Sa se implementeze clasa *Meci* (nume echipa gazda, nume echipa oaspete, scorul). Sa se construiasca o clasa *Etapa* care retine meciurile sub forma unui tablou alocat in mod dinamic. Clasa trebuie sa cuprinda metode pentru:

- Listarea tuturor meciurilor
- Afisarea meciurilor egale.

Afisarea victoriilor in deplasare.