Cerinte proiect:

* Definirea a minim 5 clase care au legatura cu cerinta proiectului
* Fiecare clasa trebuie sa contina minim un camp definit dinamic gestionat printr-un pointer si minim un vector definit static; trebuie abordate atat siruri de caractere (char\*) cat si vectori de valori numerice
* In solutie trebuie sa existe minin o clasa ce defineste un camp static si minim o clasa ce defineste un camp constant
* Toate atributele se definesc pe zona privata a clasei
* Pentru fiecare atribut se defineste o interfata publica (functii accesor) ce permite citirea si scrierea valorii; functiile de modificare trebuie sa contina minim o regula de validare a valorii
* Fiecare clasa contine minim 2 metode generice de prelucrare/afisare a atributelor
* Fiecare clasa contine minim 2 constructori cu parametri, constructor de copiere, destructor si forma supraincarcata a operatorului =
* Pentru fiecare clasa se supraincarca obligatoriu >> si <<
* Pentru fiecare clasa se supraincarca obligatoriu:
  + operatorul pentru idexare []
  + minim un operator matematic (+,-,\* sau /)
  + ++ sau -- (cu cele 2 forme)
  + operatorul cast (catre orice tip) explicit sau implicit
  + operatorul pentru negatie !
  + un operator conditional (<.>,=<,>=)
  + operatorul pentru testarea egalitatii dintre 2 obiecte ==
* Toate clasele sunt definite intr-un singur fisier sursa
* Testarea claselor (metoda main) va parcurge apelul tuturor metodelor implementate
* Se implementeaza un modul prin care aplicatia primeste fisiere de intrare (de tip text) prin argumentele functiei main (exemplu: daca aplicatia se numeste proiectPOO.exe atunci ea este lansata in executie cu comanda *proiectPOO.exe facturi.txt cheltuieli.txt;* pot fi folosite doar fisiere text)
* Se implementeaza un modul prin care utilizatorul poate salva datele aplicatiei in fisiere binare. Datele aplicatiei se considera a fi acele date obtinute din fisierele text primite ca argumente pentru main sau date incarcate de utilizatori in timpul sesiunii de lucru. Daca aplicatia nu primeste argumente prin functia main (se lanseaza in executie cu proiectPOO.exe) atunci va accesa fisiere binare aflate la locatii cunoscute de progamator. Fisirele binare permit salvarea tuturor datelor care au fost incarcate in aplicatie prin diferite metode.
* Entitatile care gestioneaza fisierele si operatiile pe fisiere se implementeaza prin clase (codul sursa prin care sunt accesate fisierele trebuie sa fie inclus in metode care sa apartina unor clase noi adaugate in proiect)
* Se implementeaza modulul prin care utilizatorul are posibilitatea sa genereze diferite rapoarte pentru toate datele (de ex. lista utilizatori, lista intrebari dintr-o anumita categorie, ierarhii utilizatori, raspunsuri pentru o intrebare, etc ) care sa fie afisate la consola dar si in fisiere text (minim 3 rapoarte)
* Aplicatia permite incarcarea de date din fisiere CSV (comma separated values);  simbolul ales pentru a separa valorile este ales de programator si este folosit pentru toate fisierele CSV
* Implementarea unei clase care sa exemplifice conceptul compunerii claselor (daca nu a fost deja implementata in fazele anterioare)
* Extinderea claselor definite in fazele anterioare prin adaugarea de atribute noi (necesare in aceasta faza) se face doar prin derivare si NU prin modificarea claselor existente - minim 2 extinderi
* Se implementeaza minim o clasa abstracta (cu sau fara atribute) ce trebuie sa continta minim 2 metode virtuale pure
* Se implementeaza minim un vector de pointeri la clasa de baza (a ierarhiei) prin care se gestioneaza elemente concrete dintr-o familie de clase;
* Sa se implementeze din libraria STL cel putin un vector, un set, o lista si un map pentru a gestiona datele aplicatiei