**21.10.2022**

**PROGRAMARE ORIENTATĂ PE OBIECTE**

**– LABORATOR NR. 3–**

1. Definiți clasa Date pentru memorarea datei sub forma zi lună an. Clasa va conţine metode de tip constructor necesare pentru următoarele definiţii de obiecte:

X`

Date d1(15, 3, 99); // zi, lună, an

Date d2(20, 4); // zi, lună, an curent

Date d3(18); // zi, lună curentă, an curent

Date d4; // zi curentă,lună curentă, an current

Pentru a extrage data currentă se pot utiliza funcții din biblioteca **ctime.** În limbajul C/C++ sunt definite 4 tipuri de date pentru manipularea unei date calendaristice: clock\_t, time\_t, size\_t, tm.

Structura tm conține câmpuri pentru a reține informații despre o data calendaristică și despre timpul local exprimat în secunde, minute, ore.

De exemplu, pentru a reține o data calendaristică se pot utiliza câmpurile: tm\_mday, tm\_mon, tm\_year.

Exemplu pentru a gestiona data currentă locală:

time\_t data=time(0) ; //numărul de secunde care au trecut de la 01.01.1970 până în momentul executării sursei

tm\* ltm=localtime(&data); // se returnează un pointer câtre o structură care reprezintă timpul local

cout<<”Anul ”<<1900 + ltm->tm\_year;

1. Se consideră un program care descrie organizarea personalului unei instituţii. Definiți tipul abstract de date **Angajat** cu datele membre **nume, varsta, salariu**. Încapsulați în clasă metode pentru inițializarea obiectelor de tip Angajat, pentru afișarea datelor membre ale unui obiect și pentru calculul salariului net. Definiți un tablou de obiecte de tip **Angajat**, datele fiecărui angajat se află în fișierul **angajat.txt**, dipuse pe câte o linie de forma nume varsta salariu.