**Proiect final: Organizarea activitatii unui bazin de inot**

Proiectul presupune realizarea unui program care să permită organizarea activității la un bazin. Pentru implementarea acestui program am gandit o modelare cu 4 clase: Bazin, Antrenor, Cursant si CursantPotential, iar pentru implementarea lor am folosit vectori.

Programul ofera următoarele funcționalități:

1. afiseaza în consolă un meniu de pornire, unde enumera si numeroteaza opțiunile de utilizare pe care le are utilizatorul. Acestea sunt: „Adaugă antrenor”, „Șterge antrenor”, „Afișează ore antrenor”, „Adaugă cursant, „Șterge cursant”, „Vizualizare cursanți potențiali” și „Închidere program”.

- implementarea meniului se face prin metoda Meniu din clasa bazin

2. dupa afisarea meniului, utilizatorul va introduce opțiunea dorită, inputul va fi validat și va fi cerut din nou dacă inputul nu corespunde uneia dintre opțiuni

3. dacă inputul e valid să șteargă ecranul și să lanseze în execuție codul corespunzător acelei

opțiuni, afisând un nou text, cu detalii despre opțiunea selectată.

4. indiferent de opțiunea selectată, meniul care se afișează să aibă și opțiunea „Revenire la ecranul anterior”. Dacă utilizatorul o selectează programul să reafișeze meniul principal.

- implementarea acestor verificari se gaseste in toate metodele unde se solicita introducerea de input

5. la selectarea opțiunii „Adaugă antrenor” va afisa textul: „Introduceți nume antrenor”, să permită utilizatorului să introducă numele si să îi atribuie un cod unic

- metoda AdaugaAntrenor este implementata in clasa Bazin, care are un membru privat de tip vector format din obiecte de tip Antrenor

- in aceasta metoda introducem numele si prenumele fiecarui antrenor si i se aloca un id unic incepand cu 1.

- introducerea antrenorului in vectorul de antrenori se face prin apelarea constructorului Antrenor, iar id-ul noului antrenor este obtinut prin aflarea id-ul maxim existent in vector si incrementarea acestuia cu 1

6. la selectarea opțiunii „Șterge antrenor”: să afișeze antrenorii deja definiți și codurile lor, să afișeze textul: „Selectați numărul antrenorului de șters”, să permită utilizatorului să introducă numărul antrenorului, să valideze inputul și să îl ceară din nou dacă inputul nu corespunde codului din dreptul unui antrenor, dacă inputul e valid să șteargă din evidență antrenorul selectat, să realoce cursanții lui către antrenorul cel mai puțin încărcat.

▪ dacă în urma realocării acesta depăsește 40 de cursanți(eu am considerat 4 cursanti) să realoce cursanți și la restul antrenorilor.

▪ dacă în urma realocării se ajunge ca toți antrenorii să aibă 40 de cursanți(am considerat 4 cursanti) și totuși mai rămân cursanți nerealocați să ofere opțiunea supraîncărcării, în mod egal, a tuturor antrenorilor.

- metoda este implementata in clasa Bazin si in prima faza se identifica antrenorul de sters

- se parcurge vectorul de antrenori pentru a identificat antrenorul ce trebuie sters, in plus retin si totalul locurilor disponibile ale celorlalti antrenori (maxim 4 cursanti poate avea fiecare)

- la parcurgere retin pozitia antrenorului de sters si pozitia antrenorului cu cei mai putini cursanti

- daca toti cursantii antrenorului de sters au loc la antrenorul cu cei mai putini cursanti atunci ii mut la acesta, dupa care sterg antrenorul de sters

- daca nu se pot realoca la un singur antrenor, parcurg vectorul de antrenori de la inceput si adaug cursantii antrenorului de sters fiecarui antrenor disponibil pana ajunge fiecare la numarul maxim de 4 cursanti (primul cursant al antrenorului de sters il adaugam la finalul vectorului de cursanti ai primului antrenor cu pozitii disponibile, dupa care stergeam primul cursant al antrenorului de sters si tot asa pana antrenorul ramane fara cursanti)

- dupa ce am finalizat toti antrenorii cu locuri disponibile, se suprascrie la toti antrenorii cate 1 cursant pana se epuizeaza toti cursantii antrenorului de sters

7. la selectarea opțiunii „Afișează cursanți antrenor”: se afișeaza o listă cu toți cursanții acelui instructor, la fiecare cursant se afișeaza informația legată de categoria lui copil, student, adult sau pensionar si se mai afișeaza și totalul cursanților pentru acel antrenor”.

- metoda este implementata in clasa Bazin

- in prima faza se selecteaza antrenorul pentru care se doreste vizualizarea cursantilor

- pentru antrenorul selectat se apeleaza metoda AfiseazaCursanti din clasa Antrenor

8. la selectarea opțiunii „Adaugă cursant”: se verifica dacă cel puțin un antrenor mai are loc în echipa sa, **dacă nu mai sunt locuri**: se afișeaza textul „Ne pare rău. Toți antrenorii noștri sunt ocupați. Lăsați-ne date de contact și vă vom contacta de îndată de apare o poziție liberă!”.

▪ să permită salvarea unui nume, a unui email și a unui număr de telefon, iar **dacă mai sunt locuri:**

▪ să afișeze textul: „Introduceți nume cursant

▪ să citească numele cursantului.

▪ în continuare să afișeze „Selectați categoria:”

1.copil

2.student

3.adult

4.pensionar

0.revenire ecran anterior

▪ să permită utilizatorului să introducă categoria, să valideze inputul și să îl ceară din nou

dacă nu e numeric și cuprins între 0 și 4

▪ dacă inputul e valid

• să șteargă ecranul

• să afișeze „Selectați antrenorul” urmat de lista antrenorilor existenți și a codurilor

lor. Lista va conține doar antrenorii care nu au deja 40 de cursanți.( 4 cursanti in varianta mea)

• să citească codul corespunzător antrenorului.

- metoda este implementata in clasa Bazin

- am adaugat intr-un un vector local de obiecte de tip Antrenor vect\_ant\_disp antrenorii unde mai erau locuri disponibile(<4)

- daca nu mai erau locuri disponibile cursantul era introdus in vectorul de obiecte CursantiPotentiali impreuna cu datele lui de identificare

- am afisat la consola antrenorii cu locuri disponibile, utilizatorul alege antrenorul la care adauga cursantul si acesta este adaugat prin metoda CreazaCursant din clasa Antrenor care apeleaza constructorul din clasa Cursant

9. la selectarea opțiunii „Șterge cursant”: se afișeaza cursanții deja definiți, se afișeaza textul: „Selectați numărul cursantului de șters”, se permite utilizatorului să introducă cursantul, se valideaza inputul și il cere din nou dacă inputul nu corespunde unui cursant, dacă inputul e valid se șterge cursantul din echipa antrenorului său si apoi se sterge cursantul.

- implementarea se face in clasa Bazin

- am afisat toti cursantii, am folosit o variabila pe care am dat-o ca si parametru metodei AfiseazaCursanti care numara cursantii deja afisati pentru antrenorul precedent

-dupa care se introduce numarul cursantului de sters, se verifica daca inputul este correct

- am parcurs vectorul de antrenori si am comparat numarul de cursanti ai fiecarui antrenor cu numarul introdus de utilizatori, daca acesta era mai mare treceam la urmatorul antrenor si numarul introdus il diminuam cu numarul de cursanti ai antrenorului precedent

10. la selectarea opțiunii „Vizualizare cursanți potențiali” -să afișeze toți cursanții potențiali care nu au găsit loc la nici un antrenor

- metoda este implementata in clasa Bazin

- se afiseaza vectorul creat anterior, in clasa AdaugaCursant, unde cursantii pentru care nu mai erau locuri dsiponbile la niciun antrenor

11. la selectarea opțiunii „Închidere program” programul se închide, programul salveaza datele despre antrenori, cursanți și cursanți potențiali în fișiere dedicate le încarca la pornire.

- implementarea este facuta in main atat pentru scrierea in fisier, cat si pentru citirea din fisier, toate datele se gasesc in Antrenori.txt, Cursanti.txt si CursantiPotentiali.txt

- am folosit streamul pentru deschiderea unui fisier in mod citire ifstream si streamul pentru scrierea in fisier ofstream

- am folosit un stringstream care permite atat scrierea cat si citirea in/din fisier, cu ajutorul carui am citit cate o linie din fiecare fisier, in fisierul Antrenori.txt fiecare linie ii corespundea unui antrenor, in fisierul Cursanti.txt pe fiecare linie erau scrisi toti cursantii antrenorului care avea aceeasi numar al liniei in fisierul de Antrenori.txt

**SERBU MIHAI-DRAGOS**

**14.10.2020**