UNIVERSITATEA “STEFAN CEL MARE”, SUCEAVA

FACULTATEA DE INGINERIE ELECTRICA SI STIINTA CALCULATOARELOR

SPECIALIZAREA CALCULATOARE

**PROIECT DISCIPLINA POO**

Autor : Gherasimescu Emanuel

**SUCEAVA . 2023**

CUPRINS

[1. Tema și motivația alegerii 2](#_Toc137068761)

[2. Capitolul 1 - Elemente teoretice 2](#_Toc137068762)

[2.1 Descrierea aplicației 2](#_Toc137068763)

[2.2 Abordarea teoretică a problemei 3](#_Toc137068764)

[2.3 Elemente specifice POO 3](#_Toc137068765)

[3.1 Tehnologii folosite 3](#_Toc137068766)

# TEMA SI MOTIVATIA ALEGERII

Tema proiectului este simularea unui joc de fotbal . Am ales această tema deoarece îmi place fotbalul, îl urmăresc și mă găndesc că acest lucru mă va motiva in desfășurarea proiectului.

# Capitolul 1 - Elemente teoretice

**2.1 Descrierea aplicației**

Aplicația ar putea fi o platformă de gestionare a competițiilor sportive sau a unui joc sportiv în care utilizatorii pot crea echipe, adăuga jucători și organiza meciuri. Fiecare jucător ar putea fi definit prin anumite caracteristici, cum ar fi nume, vârstă, poziție în teren etc. Echipele sunt fi formate dintr-o listă de jucători și ar avea opțiunea de a participa la meciuri. Meciurile ar putea avea o legătură cu două echipe participante și ar putea fi programate și gestionate în aplicație.Scopul aplicației ar putea fi de a oferi utilizatorilor o modalitate simplă și eficientă de a organiza și gestiona competiții sportive, fie pentru a juca în mod real, fie pentru a simula și gestiona meciuri virtuale.

**2.2 Abordarea teoretică a problemei**

S-au utilizat principiile de programare orientate pe obiecte POO acestea fiind dezvoltate cu ajutorul claselor, obiectelor, constructorilor de diferite tiprui. Pentru aceasta se vor utiliza clase în care se vor defini datele și metodele.

**2.3 Elemente specific POO**

Au fost definite clase precum “Player”, “Team”, “Match” , ce sunte entiăți fundamentale ale aplicației. Clasele sunt utilizate pentru a define obiecte cu attribute si comportamente specifice.

Clase:

**Player** - reprezintă un jucător și are următoarele atribute:

name (string) - numele jucătorului

age (string)- vârsta jucătorului

position (string) - poziția jucătorului în teren

goalScored (string) - numărul de goluri marcate de jucător

Metodele

publice:

Player() - constructorul implicit

Player(string Name, string Age, string Position, string GoalScored) - constructor care primește toate atributele jucătorului

string getName() const - returnează numele jucătorului

void setName() - setter pentru atributul name

void ShowData() const - afișează informațiile despre jucător (nume, vârstă, poziție, număr de goluri)

Retine informații despre un jucător, cum ar fi numele, vârsta și poziția acestuia.

**Team** - reprezintă o echipă și are următoarele atribute:

name (string) - numele echipei

coach (string) - numele antrenorului echipei

players (vector<Player>) - vector care conține jucătorii echipei

Metodele publice:

Team() - constructorul implicit

Team(string Name, string Coach) - constructor care primește numele și antrenorul echipei

string getName() - returnează numele echipei

void setCoach() - setter pentru atributul coach

string getCoach() - returnează numele antrenorului echipei

void setName() - setter pentru atributul name

void AddPlayer(const Player& player) - adaugă un jucător în echipă

void ShowData() - afișează informațiile despre echipă (nume, antrenor)

void ShowTeam() - afișează lista jucătorilor din echipă

Reprezintă o echipă și conține o listă de jucători.Metodele addPlayer(player) și removePlayer(player) permit adăugarea și eliminarea unui jucător din echipă.Metoda getPlayers() returnează lista de jucători ai echipei.

**Match** - reprezintă un meci și are următoarele atribute:

team1 (Team) - prima echipă

team2 (Team) - a doua echipă

date (string) - data meciului

time (string) - ora meciului

Metodele publice:

Match() - constructorul implicit

Match(const Team& team1, const Team& team2, string Date, string Time) - constructor care primește cele două echipe, data și ora meciului

string getDate() - returnează data meciului

string getTime() - returnează ora meciului

void setDate() - setter pentru atributul date

void setTime() - setter pentru atributul time

void ShowData() - afișează informațiile despre meci (echipele, data, ora)

void playMatch() - simulează desfășurarea meciului

# Capitolul 2 - Implementare

**3.1 Tehnologii folosite**

În implementarea programului a fost folosit limbajul de programare C++ pentru dezvoltarea aplicației.

**3.2 Funcționalități**

Introducerea datelor despre un jucător: Utilizatorul poate introduce numele, vârsta, poziția și numărul de goluri marcate de un jucător.

Adăugarea jucătorilor în echipe: Utilizatorul poate crea două echipe și poate adăuga jucătorii în fiecare echipă. Jucătorii pot fi selectați dintr-o listă de jucători disponibili sau introduși manual.

Afișarea datelor despre jucători și echipe: Utilizatorul poate afișa informații despre jucători și despre echipele create, inclusiv numele, vârsta, poziția și numărul de goluri marcate de fiecare jucător.

Organizarea unui meci: Utilizatorul poate organiza un meci între cele două echipe create. Meciul se desfășoară prin generarea aleatorie a unui scor pentru fiecare echipă și simularea atacului și apărării. Actualizarea scorului meciului: Programul actualizează scorul meciului în funcție de golurile marcate de fiecare echipă. Afișarea rezultatului meciului: Utilizatorul poate vedea rezultatul final al meciului, scorul și detalii despre echipele implicate.

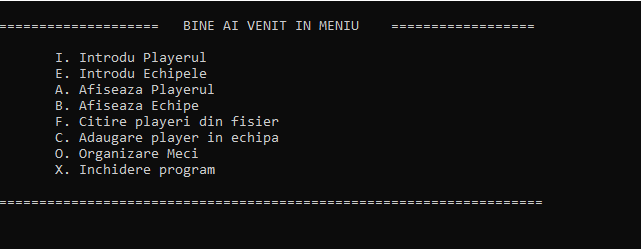
Citirea jucătorilor dintr-un fișier: Există opțiunea de a citi jucătorii dintr-un fișier text și de a-i asocia cu echipele.

**3.3 Legătura dintre clase**

Clasa Team conține un membru de tip vector, players, care stochează obiecte de tip Player. Astfel, clasa Team este asociată cu clasa Player, deoarece utilizează obiecte de tip Player în cadrul vectorului players.Clasa Match conține două obiecte de tip Team, team1 și team2. Prin intermediul acestor obiecte, clasa Match este asociată cu clasa Team.Această asociere permite clasei Match să acceseze și să interacționeze cu obiectele Team și Player, pentru a gestiona meciurile între echipe și a afișa informații relevante despre echipe și jucători.

# Analiza soluției implementate

La momentul rulării aplicației vedem că pornește cu un meniu ce are rolul de a oferi utilizatorului opțiuni și funcționalități pe care le poate selecta și utiliza în cadrul programului. Utilizatorul poate naviga și interacționa cu diverse secțiuni și acțiuni ale aplicației.



**Bibliografie**