

c++ 프로그래밍 및 실습

프로젝트 제안서

제출일자:2023.11.03

제출자명:김성준

제출자학번:182659

1. 프로젝트 목표

1) 배경 및 필요성

코로나 19 이후 , 야외 활동이 늘어나면서 축구나, 야구 등 스포츠 관람 뿐만 아니라 직접 동호회나 동아리 등 모임에 가입해서 직접 팀 스포츠에 참여 하는 비중이 늘어나고 있습니다. 프로 스포츠 클럽들과 다르게 동호회나 동아리 등 취미를 목적으로 하는 모임은 참여도가 비교적 자유로워서 경기에 참가하는 인원을 정하고 분배하는데 어려움이 있습니다. 따라서 이 과정을 자동화 하는 프로그램을 개발하는데 목표를 두었습니다.

2) 프로젝트 목표

우선 순위 큐 라는 자료구조를 이용하여 포지션 선호도나 개인 일정 등을 고려하여 경기 참여 선수 분배를 자동화 하는 '스쿼드 메이커' 라는 프로그램을 개발한다.

3) 차별점

선수 선발인원을 정하는 기존 방식은, 각 경기 일정마다 참여할 인원을 모집하고, 참여 인원이 모이면 그 인원에게 맞게 사람이 직접 해당 경기의 쿼터수를 고려하여 사람들의 개인일정, 피로도, 선호도 등을 고려해야하는 수고스러운 방식이었습니다. 따라서 기존의 방식은 시간도 오래 걸릴 뿐만 아니라 개인 사정이나 선수들의 부상과 같은 돌발 상황에 취약합니다. 또한 , 각자 선호 하는 포지션에서 플레이 하지 못하거나 공평하게 분배 되지 않을 우려가 있습니다. 따라서 이러한 과정을 자동화 하여 돌발 상황에 대처 할수 있고, 더 빠르게 사람의 수고스러움을 줄여줄수 있는 데에 차별점이 있습니다.

2. 기능 계획

1) 선택 메뉴 출력 기능

- 사용자의 요구에 따라 원하는 메뉴를 출력하여 보여주거나 입력을 저장합니다.

(1) 선택 메뉴 구성

1. 선수 정보 입력/수정/삭제 항목
2. 포메이션 정보 입력/수정/삭제 항목
3. '스쿼드메이커' 항목

2) 선수 정보 입력/수정/삭제 항목 기능

- 선수의 이름, 선호 포지션/비선호 포지션, 등번호를 저장하는 항목입니다.

3) 포메이션 정보 입력/수정/삭제 항목 기능

- 경기에서 사용할 포메이션을 저장하는 항목입니다. 주로 팀스포츠는 인원에 따라, 전술에 따라 각 자리의 역할이 달라집니다. 축구를 예로 들자면, 포메이션에 따라 수비수나 미드필더 공격수의 숫자가 달라집니다. 5-2-3 포메이션의 경우 공격-중앙-수비에 각각 5명·2명·3명을, 4-3-3 포메이션의 경우에는 4명·3명·3명을 배치합니다.

4) 스쿼드 메이커 기능

- 입력된 선수정보, 포메이션 정보를 바탕으로 대전방식/참가자수/포메이션/쿼터 수/개인사정 등의 사용자의 입력을 받아 우선순위 큐를 바탕으로 각 쿼터에 인원을 배치합니다.

예시) 다음은 실제 경기에서 사용하기 위해 기존의 방식으로 작성한

14명의 참여 인원/ 11명 선발인원/ 4쿼터를 고려하여 작성한 결과물로, 이번 프로젝트의 예상 되는 결과물 입니다.

<p>①</p> <p>○ 화석 ○ 민호 ○ 장우</p> <p> ○ 성백</p> <p> ○ 명재 ○ 승민</p> <p>○ 유민 ○ 장민 ○ 채 ○ 신윤</p>	<p>②</p> <p>○ 화석 ○ 민호 ○ 아넷 (Janek)</p> <p> ○ 성백</p> <p> ○ 명재 ○ 하민</p> <p>○ 정민 ○ 중민 ○ 권희 ○ 병남</p>	<p>②</p> <p>○ 화석 ○ 민호 ○ 아넷 (Janek)</p> <p> ○ 성백</p> <p> ○ 명재 ○ 승민</p> <p> ○ 장민 ○ 하민</p> <p>○ 재홍 ○ 중민 ○ 권희 ○ 임우</p> <p> ○ 능환 ○ 정민 ○ 권희 ○ 병남</p> <p>④</p> <p>1. ○ 장민 ○ 민호 ○ 아넷 (Janek)</p> <p> ○ 명재</p> <p> ○ 장민 ○ 하민</p> <p> ○ 정민 ○ 권희 ○ 병남</p>
-----------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

세부기능 1) 선택 메뉴를 통해 사용자의 입력 받기

ex) 참여인원/ 선발인원/ 쿼터수/대전방식(내전,대전) / 개인 사정

세부기능 2) 기존의 정보와 사용자 입력을 바탕으로 우선순위를 사용하여 결과물 출력하기

3. 진척사항

1) 기능 구현

기능 1.선택메뉴 출력

```
#include <windows.h> // 콘솔창 clear 및 시간 지연을 주기위한 헤더파일

#include <iostream>
#include <vector>

#include "Player.h"

using namespace std;

// system("cls"); //콘솔창 clear하는 명령어

// 함수 선언
void DisplayMenu(); //사용자의 요구에 따라 원하는 메뉴를 출력
void loadPlayersInfo(vector<Player> &players); // 기존 사용자 정보 불러오기
void PlayerMenu();

vector<Player> players; //

int main() {
    cout << "-----스쿼드 메이커 프로그램을 시작합니다.----- \n\n";
    cout << "-----기존 정보를 불러옵니다-----\n\n";

    loadPlayersInfo(players); // 기존 정보 불러오기

    DisplayMenu(); //메인 메뉴 출력

    return 0;
}

}
```

- 입출력
 - user_input : 사용자 입력
 - vector<Player> players : 플레이어 정보를 담은 객체 배열
- 사용된 함수
 - Displaymenu()

```
void DisplayMenu() {
    //메인메뉴 출력
    cout << "메인 메뉴입니다.\n\n";
    cout << "\n-----\n";
    cout << "항목을 선택해주세요.\n\n";
    cout << "0. 종료\n";
    cout << "1. 선수 정보 입력/수정/삭제\n";
    cout << "2. 포메이션 정보 입력/수정/삭제\n";
    cout << "3. 스쿼드메이커\n";
    cout << "\n-----\n";
    cout << "사용자 입력:";
    int user_input;
    cin >> user_input;
    system("cls"); //콘솔창 clear
    switch (user_input) {
        case 0:
            cout << "프로그램을 종료합니다.\n";
            exit(0); //프로그램 종료
            break;
        case 1:
            PlayerMenu();
            break;
        case 2:
            cout << "포메이션 정보 입력/수정/삭제 화면 입니다.\n";

            // 구현 예정
            cout << "아직 기능이 구현되지 않았습니다. 메인메뉴로 돌아갑니다.\n";
            DisplayMenu();
            //
    }
```

```

        break;
    case 3:
        cout << "스쿼드메이커 화면 입니다.\n";
        // 구현 예정
        cout << "아직 기능이 구현되지 않았습니다. 메인메뉴로 돌아갑니다.\n";
        DisplayMenu();
        //
        break;
    default:
        cout << "잘못된 입력입니다. 다시 선택 해주세요 .\n\n";
        DisplayMenu();
    }
}

```

■ 함수 입출력:

- user_input : 사용자 입력 저장

■ 함수 설명 :

- 메뉴 항목을 출력하고 입력을 switch 조건문을 이용해 구분한다.
 - 0은 종료
 - 1~3의 입력의 경우 각 항목을 호출한다.

◦ loadPlayersInfo(vector<Player> &players) 함수

```

void loadPlayersInfo(vector<Player> &players) {
    // 저장된 플레이어 정보
    players.push_back(Player("김건휘", "공격", "골키퍼"));
    players.push_back(Player("박민호", "공격", "수비"));
    players.push_back(Player("아넷", "공격", "수비"));
    players.push_back(Player("다윗", "중량", "수비"));
    players.push_back(Player("송민", "중량", "수비"));
    players.push_back(Player("종훈", "수비", "없음"));
    players.push_back(Player("희석", "수비", "공격"));
    players.push_back(Player("준희", "수비", "공격"));
    players.push_back(Player("영훈", "골키퍼", "없음"));

    Sleep(1500);
    cout << "플레이어 정보 로드 완료\n\n";
    cout << "-----\n";
    Sleep(1000);
    system("cls");
}

```

■ 함수 입출력 :

- vector<Player> &players : player 객체 배열

■ 함수 설명 :

- push_back()함수를 통해 객체배열에 저장된 정보를 입력한다.
- Sleep()함수를 통해 시각적인 효과를 추가하여 각각 1.5초,1초의 지연을 추가한다.
- 플레이어 정보 로드 1초 이후 system("cls") 를 통해 콘솔창을 초기화 한다.

• 기능 설명

- 함수 시작과 함께 각 함수를 호출하여 기존 정보를 업로드하고,선택메뉴를 출력한다.
- 메인 메뉴의 선택입력이 0 인경우 프로그램이 종료된다.
- 선택입력이 1~3인경우 각 메뉴의 화면을 출력하도록 함수를 호출하고, 각 작업이 끝나면 사용자의 입력을 받아 메인 메뉴를 호출 하거나 작업을 계속한다.
- 사용자의 입력이 정상입력이 아닌 경우 다시 입력을 받도록 함수를 다시 호출한다.

- 적용된 배운 내용: 객체 배열 , 함수, switch,

기능2. 선수정보 입력/수정/삭제 기능

```

void PlayerMenu() {
    cout << "\n\n선수 정보 입력/수정/삭제 화면 입니다. \n\n";
    Player player;
}

```

```

player.GetInformation(players); //현재 플레이어 정보 출력
cout << "메뉴를 선택해주세요.\n\n";
cout << "0.메인으로 돌아가기 \n";
cout << "1.선수 정보 입력\n";
cout << "2.선수 정보 수정\n";
cout << "3.선수 정보 삭제\n";
cout << "\n-----\n";
cout << "사용자 입력:";
int user_input;
cin >> user_input;
switch (user_input) {
    case 0:
        DisplayMenu(); //메인 메뉴로 돌아갑니다
        break;
    case 1:
        player.InsertPlayer(players);
        PlayerMenu();
        break;
    case 2:
        player.EditPlayer(players);
        PlayerMenu();
        break;
    case 3:
        player.DeletePlayer(players);
        PlayerMenu();
        break;
    default:
        cout << "잘못된 입력입니다.다시 선택 해주세요 .\n\n";
        PlayerMenu();
}

```

- 입출력 :
 - Player player:빈객체
 - players: 플레이어 정보가 저장된 객체 배열
 - user_input: 사용자 입력 저장
- 기능 설명:
 - 메인 메뉴를 통해 PlayerMenu() 함수를 호출하여 기능한다.
 - 기본적인 플레이어 메뉴 항목을 출력하고, 사용자 입력을 switch 조건문을 이용해 구분한다.
 - 빈 객체를 이용하여 Player클래스(추후 설명)의 멤버함수를 이용한다.
 - 1~3은 각각 멤버함수를 이용하여 입력/수정/삭제를 시행하고,0 입력시 메인메뉴로 돌아간다. 그 외의 입력이 들어온 경우 오류 메시지를 출력하고 다시 입력받는다.

2) 클래스 및 헤더파일 구현

- Player 클래스& 헤더파일
 - 헤더파일

```

#pragma once

#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Player {
public:
    string name; //플레이어 이름
    string pre_pos; //선호 포지션
    string non_pre_pos; //비선호 포지션
    Player(); //기본 생성자
    Player(string name, string pre_pos, string non_pre_pos);
    void InsertPlayer(vector<Player> &players); // 플레이어 추가
    void DeletePlayer(vector<Player> &players); //플레이어 삭제
    void EditPlayer(vector<Player> &players); // 플레이어 수정
    void GetInformation(const vector<Player> &players); // 플레이어 정보 보여주는 함수
    string GetValidPosition(const string &prompt); // 입력된 포지션 유효성 검사 및 올바른 입력 반환하는 함수
};

```

- 클래스

```

#include "Player.h"

#include <iomanip> //setw()사용하기 위함
#include <iostream>
#include <string>
#include <vector>

// 생성자 정의와 초기화 목록 사용
Player::Player() {}
Player::Player(string name, string pre_pos, string non_pre_pos) {
    this->name = name;
    this->pre_pos = pre_pos;
    this->non_pre_pos = non_pre_pos;
}

// 다른 멤버 함수들의 구현은 여기에 추가
void Player::InsertPlayer(vector<Player> &players) {
    cout << "-----\n";
    cout << "플레이어를 추가합니다.\n\n";
    cout << "플레이어의 정보를 이름/선호 포지션/비선호 포지션 순으로 "
        "입력하세요.\n\n";
    cout << "선호/비선호 포지션은 공격/중양/수비/골키퍼/없음 중에서 "
        "입력해주세요\n\n";
    Player input;
    // 이름 입력
    cout << "이름: ";
    cin >> input.name;

    // 선호 포지션 입력 및 유효성 검사
    input.pre_pos = GetValidPosition("선호 포지션: ");
    input.non_pre_pos = GetValidPosition("비선호 포지션: ");
    players.push_back(input);
    cout << "-----\n";
    cout << "선수가 추가되었습니다.\n\n";
    cout << "-----\n";
}

void Player::DeletePlayer(vector<Player> &players) {
    cout << "-----\n";
    cout << "플레이어를 삭제합니다.\n\n";
    cout << "삭제할 플레이어의 번호를 입력해주세요 \n\n";

    int delete_index; // 삭제할 플레이어 번호

    // 삭제할 번호 입력
    cout << "번호: ";
    cin >> delete_index;
    players.erase(players.begin() + delete_index - 1);

    cout << "선수가 삭제 되었습니다.\n";
}

void Player::EditPlayer(vector<Player> &players) {
    cout << "-----\n";
    cout << "플레이어를 수정합니다.\n\n";
    cout << "수정할 플레이어의 번호를 입력해주세요 \n\n";

    int edit_index; //수정할 플레이어 번호

    // 수정할 플레이어 번호 입력
    cout << "수정할 플레이어 번호: ";
    cin >> edit_index;
    cout << "플레이어의 정보를 이름/선호 포지션/비선호 포지션 순으로 "
        "입력하세요.\n\n";
    cout << "선호/비선호 포지션은 공격/중양/수비/골키퍼/없음 중에서 "
        "입력해주세요\n\n";
    cout << "이름: ";
    cin >> players[edit_index - 1].name;
    players[edit_index - 1].pre_pos = GetValidPosition("선호 포지션:");
    players[edit_index - 1].non_pre_pos = GetValidPosition("비선호 포지션:");

    cout << "선수정보가 수정 되었습니다.\n";
}

void Player::GetInformation(const vector<Player> &players) {
    cout << "번호 | 이름 | 선호 포지션 | 비선호 포지션\n";
    cout << "-----\n";

    for (int i = 0; i < players.size(); i++) {
        cout << setw(4) << i + 1;
        cout << " | " << setw(19) << players[i].name;
        cout << " | " << setw(16) << players[i].pre_pos;
        cout << " | " << players[i].non_pre_pos << endl;
    }
}

```



```

    }
    cout << endl;
    cout << "-----\n";
}

string Player::GetValidPosition(const string &prompt) {
    string position; // 입력을 통해 들어온 포지션
    while (true) {
        cout << prompt;
        cin >> position;

        if (position == "공격" || position == "중양" || position == "수비" ||
            position == "골키퍼" || position == "없음") {
            break;
        } else {
            cout << "잘못된 포지션입니다. 공격/중양/수비/골키퍼/없음 중 하나를 "
                "입력하세요.\n";
        }
    }
    return position;
}

```

- 클래스 설명

- 멤버변수 :

- string name; //플레이어 이름
- string pre_pos; //선호 포지션
- string non_pre_pos; //비선호 포지션

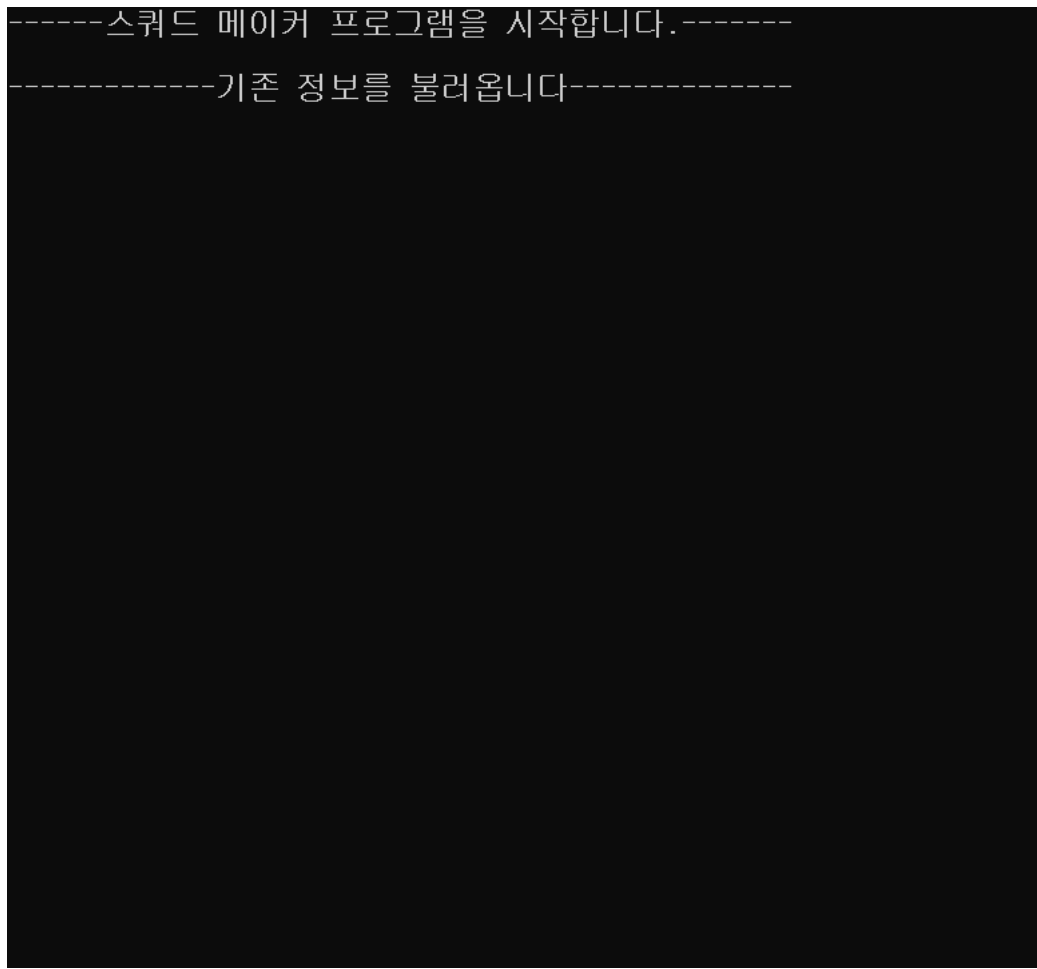
- 멤버함수설명 :

- void InsertPlayer(vector<Player> &players)
 - 입력 조건을 출력한다.
 - 사용자 입력 받기 : 이름 과 달리 선호/비선호 포지션은 사용자의 입력이 정상 입력인지 유효성 검사를 하는 함수를 호출한다.(유효할때 까지 입력 받는다.)
 - push_back()를 통해 객체 배열에 추가하고 알림문구를 출력한다.
- void DeletePlayer(vector<Player> &players); //플레이어 삭제
 - 삭제할 플레이어의 번호를 입력받고, erase()함수를 이용해 해당 인덱스의 객체를 제거한다.
- void EditPlayer(vector<Player> &players); // 플레이어 수정
 - 수정할 플레이어의 번호를 입력받고, 해당 인덱스의 객체 정보를 덮어쓰기 한다. 덮어쓰기 하는 과정은 선수 입력 과정 과 동일하게 유효성 검사를 한다.
- void GetInformation(const vector<Player> &players); // 플레이어 정보 보여주는 함수
 - 객체배열을 파라미터로 입력 받고, 저장된 값을 출력한다.
 - setw()함수를 이용하여 출력화면에 고르게 정렬되도록 한다.
- string GetValidPosition(const string &prompt); // 입력된 포지션 유효성 검사 및 올바른 입력 반환하는 함수
 - prompt : 값을 달리하여 선호/비선호 포지션 입력시 재사용 할수 있도록 사용자 입력 칸에 출력되는 값을 저장한다.
 - 공격/중양/수비/없음 중의 입력이 아닌경우 오류 메시지를 출력하고 다시 입력 받는다.
 - 유효성 검사를 통과한 포지션 값을 반환한다.

2) 테스트 결과

(1) 선택메뉴 출력

세부기능1: 저장된 정보 출력



세부기능2: 메인메뉴 출력

메인 메뉴입니다.

항목을 선택해주세요.

- 0. 종료
- 1. 선수 정보 입력/수정/삭제
- 2. 포메이션 정보 입력/수정/삭제
- 3. 스쿼드메이커

사용자 입력:

선수정보 입력/수정/삭제 화면

선수 정보 입력/수정/삭제 화면 입니다.

번호	이름	선호	포지션	비선호	포지션
1	김건호	골키퍼	공격수	골키퍼	공격수
2	박민아	수비수	수비수	수비수	수비수
3	최윤희	수비수	수비수	수비수	수비수
4	김민석	수비수	수비수	수비수	수비수
5	최민준	수비수	수비수	수비수	수비수
6	최민준	수비수	수비수	수비수	수비수
7	최민준	수비수	수비수	수비수	수비수
8	최민준	수비수	수비수	수비수	수비수
9	최민준	수비수	수비수	수비수	수비수

메뉴를 선택해주세요.

- 0.메인으로 돌아가기
- 1.선수 정보 입력
- 2.선수 정보 수정
- 3.선수 정보 삭제

사용자 입력:1

메인메뉴 되돌아가기

메뉴를 선택해주세요.

- 0.메인으로 돌아가기
- 1.선수 정보 입력
- 2.선수 정보 수정
- 3.선수 정보 삭제

사용자 입력:0
메인 메뉴입니다.

항목을 선택해주세요.

- 0.종료
- 1.선수 정보 입력/수정/삭제
- 2.포메이션 정보 입력/수정/삭제
- 3.스쿼드메이커

사용자 입력:

(2) 선수 정보 입력/수정/삭제 항목 기능

- 메인화면

선수 정보 입력/수정/삭제 화면 입니다.

번호	이름	선호 포지션	비선호 포지션
1	김건호	공격	골키퍼
2	박민호	공격	수비
3	박민호	공격	수비
4	박민호	공격	수비
5	박민호	공격	수비
6	박민호	공격	수비
7	박민호	공격	수비
8	박민호	공격	수비
9	박민호	공격	수비

메뉴를 선택해주세요.

0.메인으로 돌아가기
 1.선수 정보 입력
 2.선수 정보 수정
 3.선수 정보 삭제

사용자 입력:1

- 플레이어 추가

사용자 입력:1

플레이어를 추가합니다.

플레이어의 정보를 이름/선호 포지션/비선호 포지션 순으로 입력하세요.
 선호/비선호 포지션은 공격/중앙/수비/골키퍼/없음 중에서 입력해주세요

이름: 김성준
 선호 포지션: 중앙
 잘못된 포지션입니다. 공격/중앙/수비/골키퍼/없음 중 하나를 입력하세요.
 선호 포지션: 중앙
 비선호 포지션: 있음
 잘못된 포지션입니다. 공격/중앙/수비/골키퍼/없음 중 하나를 입력하세요.
 비선호 포지션: 없음

선수가 추가되었습니다.

- 플레이어 수정

선수 정보: 중앙
비선수 정보: 수비
선수 정보가 수정 되었습니다.

선수 정보 입력/수정/삭제 화면 입니다.

번호	이름	선수 포지션	비선수 포지션
1	김건호	공격수	골키퍼
2	박민호	공격수	수비수
3	박민호	공격수	수비수
4	박민호	공격수	수비수
5	박민호	공격수	수비수
6	박민호	공격수	수비수
7	박민호	공격수	수비수
8	박민호	공격수	수비수
9	박민호	공격수	수비수
10	박민호	공격수	수비수

메뉴를 선택해주세요.

- 0.메인으로 돌아가기
- 1.선수 정보 입력
- 2.선수 정보 수정
- 3.선수 정보 삭제

사용자 입력:

- 플레이어 삭제

삭제할 플레이어의 번호를 입력해주세요

번호: 10
선수가 삭제 되었습니다.

선수 정보 입력/수정/삭제 화면 입니다.

번호	이름	선호 포지션	비선호 포지션
1	김건희	공격수	골키퍼
2	박민호	공격수	수비수
3	박민호	공격수	수비수
4	박민호	공격수	수비수
5	박민호	공격수	수비수
6	박민호	공격수	수비수
7	박민호	공격수	수비수
8	박민호	공격수	수비수
9	박민호	공격수	수비수

메뉴를 선택해주세요.

0.메인으로 돌아가기
1.선수 정보 입력
2.선수 정보 수정
3.선수 정보 삭제

사용자 입력:

4. 계획 대비 변경 사항

1) 선택메뉴 출력 기능 (내용 추가)

- 추가내용: 저장된 정보를 불러오는 함수를 추가하였습니다. 추후 포메이션 클래스 추가여부에 따라 기능을 분리할 예정입니다.
- 사유
 - 테스트 상황 뿐만 아니라, 사용자가 이 프로그램을 사용한다고 생각했을때 반복되는 작업을 한다면 매우 번거로울 것으로 판단되어, 프로그램에 정보를 내장할수 있도록 하였습니다.

2) Player클래스 &헤더 추가

- 사유 : 객체 배열을 통해 객체를 쉽게 다루고, 작업 편리성을 위해 헤더파일에는 변수및 함수의 정의를 작성하였고, 구현부분을 분리하였습니다.

5. 프로젝트 일정 (간트차트)

일정	소요일
제안서 작성	3일
기능1(메뉴 출력)	2일
기능2(선수 정보)	2일
기능3(포메이션 정보)	3일
기능4_1(사용자 입력 받기)	3일(기능 1-2-3 선행)
기능4_2(최종 결과 출력)	5일 (기능 1~4_1 선행)
기능5-1Player 클래스	3일
기능5-2Formation클래스	3일

업무	11/3	11/10	11/17	11/26	12/03	12/10	12/12
----	------	-------	-------	-------	-------	-------	-------

제안서 작성	----->						
기능1			---->				
기능2				---->			
기능3					--	--->	
기능4	세부기능 1					----->	
	세부기능 2					----	--->
기능4	세부기능 1			---->			
	세부기능 2				---->		

- 위의 간트차트는 예비·여유 기간을 고려하여 간이로 작성하였으며, 기능 추가/삭제 여부에 따라 업데이트 될 예정입니다.