EOOP - final project

Date: 23/04/2022 **Semester: Spring 2022**

Author and Group:Chen Song; Group 102

Subject (Keyword):Tennis Club

I. Description of the project

1. Overview of the project

This project is a tennis club manage system and it designed for tennis club manager, aim to select good tennis players in the club, a tennis club has countable courts can buy and sell, and this project can also add, kick, find any members(player or coach) in this club, the information of any member can be modified as well.
One noticeable function of this project is about 'level', system will automatically allocate each player their salary and corresponding 'level' according to their win-rate.
Each coach will automatically allocate their 'salary' and 'qualification' according to their 'experience'.

2. Class and data structures overview

There are the following classes: Court, Coach, Player, Club;

class Club

```
#pragma once
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<string>
#include"default_values.hpp"
#include"coach.hpp"
#include"court.hpp"
#include"player.hpp"
using namespace std;
class Club
{
private:
    list<Court*>allCourts;//court pointer list in the club
    list<Coach*>allCoachs;//coach pointer list in the club
    list<Player*>allPlayers;//player pointer list in the club
    string name, location;//club name and location
    //string* activity;//to be considered
    int maxCourt;//the number of court is limited by club space
    int id;//club id number
public:
    Club(int courtNumber, string name, string location, int id) :maxCourt(courtNumber), name(name), location(location), id(id) {};//count number 1-5
     ~Club();//****
    //Check if there exists input player and coach, if not exist, an error message will be printed

    void AssignPlayerToCoach(int playerId, int coachId);
```

```
25.       void AssignCoachToCourt(int coachId, int courtId);
26.       //delete pervious pointer and call assign method
27.       void ReassignPlayerToCoach(int playerId, int coachId);
28.       void ReassignCoachToCourt(int coachId, int courtId);
29.
30.       //if input player id corresponds no player an error message will be printed
31.       //if player has assigned to a coach or a court, coach id and court will be printed
32.       //if player id is valid will no assignment, function will return true
33.       bool CheckPlayerValidity(int id);
34.       //if input coach id corresponds no coach an error message will be printed
35.       //if coach has been assigned to a court, true case will be returned
36.       bool CheckCoachValidity(int id);
37.       //if input court id corresponds no court an error message will be printed
38.       bool CheckCourtValidity(int id);
39.       //check if court id exists or not at first, and check if the number of court exceed max court
    number
40.       //then check input player number and coach number is within the max range or not
41.       //if not, an error message will be printed
42.       void AddCourt(int Id, char type, char size);
43.       void AddPlayer(string name, int id, int experience, float winRate);
44.       //according to given experience value automatic assign current coach's salary and level
45.       void AddCoach(string name, int id, int experience);
46.       //check if there exists a court with the input id
47.       //if not exist an error message will be printed
48.       //if this court's player list or coach list is not empty, a notice will be printed to user to
    reassign this court's player list and coach list to another court
49.       //if this court's player list and coach list are empty, delete this court and the point in the
     court list
50.       void RemoveCourt(int id);
51.       //check if there exists a coach with input id
52.       //if not exist an error message will be printed
53.       //if exists then check it's player list is empty or not
54.       //if this coach's player list is empty, delete this coach and the pointer in the coach list an
    d court coach list
55.       //if not exist an notice will be printed to the user to reassign this coach's player list to a
    nother coach
56.       void RemoveCoach(int id);
57.       //check if there exists a player with input id
58.       //if not exist an error message will be printed
59.       //if exists then delete this player and the pointer
60.       void RemovePlayer(int id);
61.       //print all coaches with input court id
62.       //if given court it is not valid or empty, an error message will be printed
63.       void PrintCoach(int courtId);
64.       //print all players with input coach id
65.       void PrintPlayer(int coachId);
66.       void PrintClub(int clubId);
67.       //print all court in the club
68.       void printCourtList();
69.       //first check if there exist the court of input court id
70.       //if not exist an error message will be printed
71.       //if exist, input court's all coach will be printed
72.       void printCoachList();
73.       //firtst check if there exist the coach of input coach id
74.       //if not exist an error message will be printed
```

```
        //if exist, input coach's all player will be printed
        void printPlayerList();
        //if input id corresponds with no coach, an error message will be printed
    //if input id is valid, the function will check this coach's experience, and upgrade his level with corrsponding experience, the salary will be modified as well
        void updateCoachData(int coachId);
        void updatePlayerData(int playerId);
        //get the number of courts in the club
        //this method is used for condition check
        size_t getCourtNumber();
        //get the number of coaches in the club
        size_t getCoachNumber();
        //get the number of players in the club
        size_t getPlayerNumber();
        //interface is used to interact with user with obvious instructions
        //void interface();
        //print basic interface information
        //void showMenu();
        //find coach assigned
        Court* findCoachAssigned(int coachId);
        //find player assigned
        Coach* findPlayerAssigned(int playerId);
    };
```

class Court

```
    class Court {
    private:
        //some other data
        int courtId;
        //court type can only be given 'I' or 'O', which means indoor and outdoor respectively
        //court size can only be given 'L', 'M' or 'S', which means large size, middle size and small size respectively
        char courtType, courtSize;
        //list<Coach*>courtCoaches;//court coachs
        int maxCoach;//max coach number (most 2)
    public:
        //set court with coachs and players
        Court(int Id, char type, char size);
        ~Court();
        //link pointer between designated coach and court
        void linkCoach(Coach& coach);
        //delete pointer between court and designated coach
        void releaseCoach(Coach& coach);
        int getId();
        //print this court's all coach, if there is no coach, an error message will be printed
        //void printCoachList();
        //used for condition check: if the coach number exceed the max court number(2) or not
        size_t getCoachNumber();
        //get coach list in order to be used in remove court function
        //list<Coach*>getCourtCoaches();
        //print court information
        void printCourt();
        //print coach list in this court
        void printCoachList();
        //hire a coach to court
```

```
        void hireCoach(int coachId);
        //fire all coaches in this court
        //required to remove all coaches while removing a not empty court
        //the reason why here require to input court id, is this need to be compared in further condition
        //void fireAllCoach(int id);
        //check if coach is exist in the court or not, if exists return true, else return false
        bool checkCoachInCourt(int coachId);
    };
```

class Coach

```
    class Coach
    {
    private:
        string name;
        int salary, id, experience;
        char qualification;//char type can only choose'A','B','C','D'
        //list<Player*>coachPlayers;
        int maxPlayer;//max player number (most 8)
    public:
        //constructor
        Coach(string name, int id, int experience) :name(name), id(id), experience(experience) {
            this->maxPlayer = MaxCoachPlayerNumber;
        }
        ~Coach();
        //add the pointer between this coach and designated player
        void linkPlayer(Player& player);
        //delete the pointer between this coach and designated player
        void releasePlayer(Player& player);
        int getId();
        //void printPlayerList();
        size_t getPlayerNumber();
        //for update use
        int getExperience();

        //get coach list in order to be used in remove court function
        //list<Player*>getCoachPlayers();
        //used for modify information in club class
        void modifyQualification(char qualification);
        //used for modify information in club class
        void modifySalary(char qualification);
        void srandSeed();

        //print this coach's information
        void printCoach();
        //print player list of this coach
        void printPlayerList();
        //hire a player of this coach
        void hirePlayer(int playerId);
        //fire a player of this coach
        //void firePlayer(int playerId);
        //check if coach is exist in the court or not, if exists return true, else return false
        bool checkPlayerInCoach(int playerId);
    };
```

class Player

```
class Player {
private:
    string name;
    int id;
    int transferFee;
    int experience;
    int salary;
    float winRate;
    char level;
public:
    //constructor with name,id,transfer fee, experience, salary, win rate and level
    Player(string name, int id, int experience, float winRate)
        :name(name), id(id), experience(experience), winRate(winRate) {};
    ~Player();

    //print this player's information
    void printPlayer();

    //for update use
    int getId();
    int getExperience();
    float getWinRate();
    //used for modify information in club class
    void modifyLevel(char level);
    void modifySalary(char level);
    void modifyTransferFee(char level);
    void srandSeed();
};
```

3. Restrictions, limits, assumptions

```
#define MaxCoachPlayerNumber 4
#define MaxCourtNumber 5
//designed max player number is 80, but here use 10 in order to easier test the limit case
#define MaxPlayerNumber 10
#define MaxCoachNumber 10
#define MaxCourtCoachNumber 2
```

Restrictions in 'Player' class:

1. Player salary, transfer fee should be an integer: salary(2000-90000), transfer fee(5000-200000), level and salary will be allocated automatically,

'D' level: win-rate: under 35, salary (2000-4000), transfer fee: 5000-10000

'C' level: win-rate: 35-50, salary(4000-8000), transfer fee: 10000-30000

'B' level: win-rate: 50-70, salary(8000-30000), transfer fee: 30000-80000

'A' level: win-rate: more than 70, salary(30000-90000), transfer fee: 80000-200000

2. Each player has only one coach
3. member level will determines his/her type, there are 4 levels(A,B,C,D), player level reached C or D is call 'normal player', reached A or B called 'professional player'.
4. Player win rate should be positive float value(0-100);

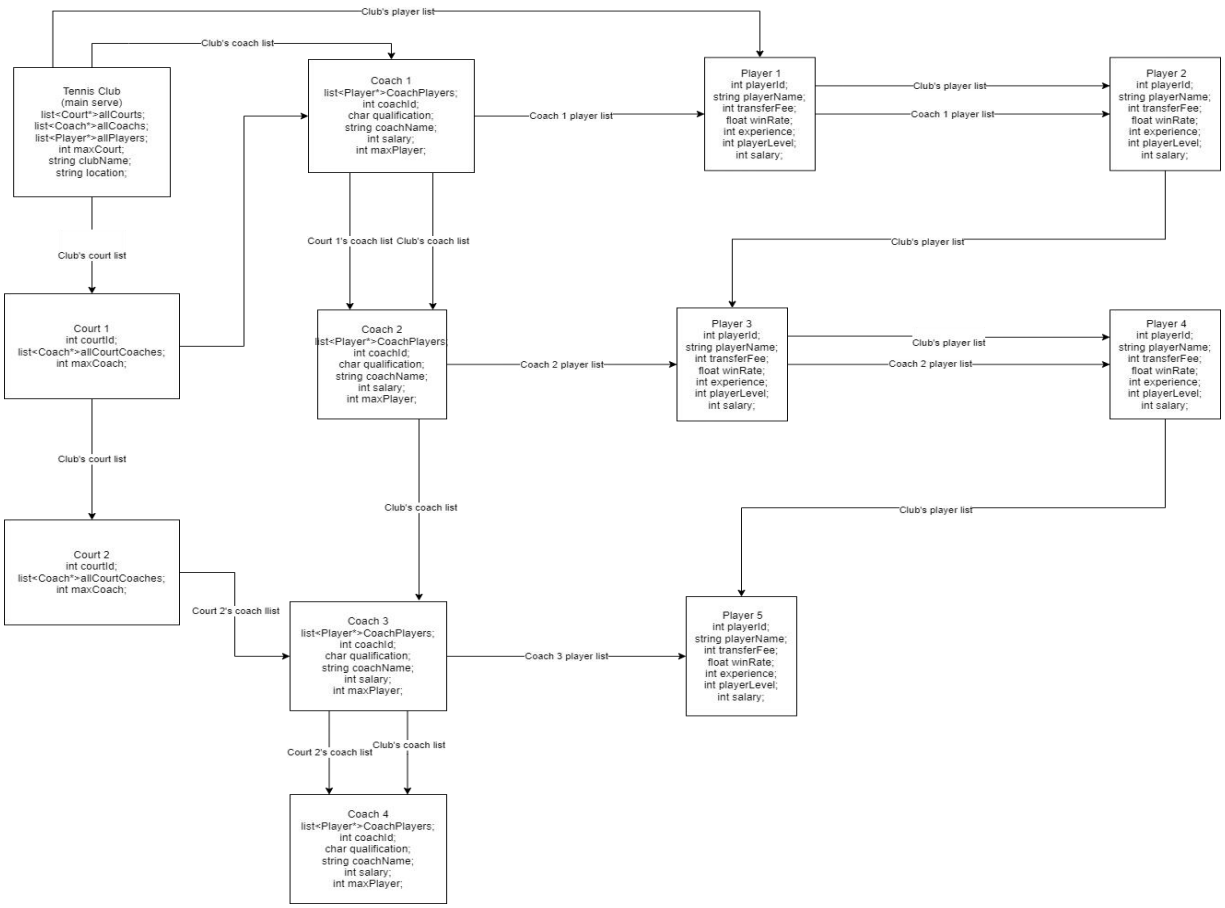
Restrictions in 'Coach' class:

1. Coach salary should be an integer: salary(2000-110000), their salary and qualification will be given automatically in the system.
1 year experience with 'D' qualification, salary (2000-10000)
2 year experience with 'C' qualification, salary(10000-20000)
3 year experience with 'B' qualification, salary(20000-40000)
4 year experience with 'A' qualification, salary(40000-70000)
2. One coach can only train 4 players and located in one court;(which means 1 court can with 2 coach but 1 coach can only in 1 court);

Restrictions in 'Club' class:
1.Club can accommodate at most 80 members and 10 coaches and contain at most 5 courts;
2.'tennis court' number is an integer with the range(1-5);

Restrictions in 'Court' class:
1. One court can contain at most 2 coaches, and each court has their own id.
2. `court type can only be given 'I' or 'O', which means indoor and outdoor respectively`
3. `court size can only be given 'L', 'M' or 'S', which means large size, middle size and small size respecti vely`

II. Case study (a memory map)



One club has three lists(court list, coach list and player list), one court has one coach list, and one coach has their player list.

III. Functional test cases

```
//-------------------------------------------------------------------------------------------------------------
//test function in class club
//Check method in 'Club'(AddCourt, AddCoach and AddPlayer)
//Testing for create a club, add a court, a coach and a student
//check the number of court, coach and player is 1 or not(to check if the add is success or not)
//fail case: 'add failed' will be shown on the screen
void test01() {
    cout << "test1: test for create a club" << endl;
    Club club(1, "Tennis Club", "Zloty Tarsy", 01);
    club.AddCourt(01,'I','L');
    club.AddCoach("Jiaxu Luo", 01, 1);
    club.AddPlayer("Haolin Li", 01, 1, 50.0);

    if (club.getCoachNumber() == 1 && club.getCourtNumber() == 1 && club.getPlayerNumber() == 1) {

        cout << "add successful!" << endl;
        cout << "Coach number:" << club.getCoachNumber() << " " << "court number: " << club.getCourtNumber() << " "
            << "Player number: " << club.getPlayerNumber() << " " << endl;
    }
    else {
        cout << "add failed!" << endl;
    }
}
```

```
test1: test for create a club
current court number: 1
Current coach number: 1
Current player number: 1
add successful!
Coach number:1 court number: 1 Player number: 1
请按任意键继续. . .
```

```
//test2: print the existing lists of court, coach and player
//Check method in 'Club'(printCourtList, printCoachList and printPlayerList)
//test for printing club information with correct input id case
//test for print club information with wrong input id case
void test02() {
    cout << "test2: print the existing lists of court, coach and player:" << endl;
    Club club(1, "Tennis Club", "Zloty Tarsy", 01);
    club.AddCourt(01,'I','L');
    club.AddCoach("Jiaxu Luo", 01, 1);
    club.AddPlayer("Haolin Li", 01, 1, 50.0);

    club.printCourtList();
    club.printCoachList();
    club.printPlayerList();
    cout << "test for print club information: " << endl;
    club.PrintClub(1);
    cout << "test for print club wrong case(print not existing member in the club): " << endl;
    club.PrintClub(2);
}
```

```
test2: print the existing lists of court, coach and player:
current court number: 1
Current coach number: 1
Current player number: 1
Court id: 1  type: Indoor  size: Large
Coach name: Jiaxu Luo  id: 1  salary: 9133  experience: 1  qualification: D
Player name: Haolin Li  id: 1  transfer fee: 51133  experience: 1  salary: 29133  win-rate: 50%  level: B
test for print club information:
Club name: Tennis Club  id:1   max court number: 1   location: Zloty Tarsy
test for print club wrong case(print not existing member in the club):
Input club id corresponding no club!
请按任意键继续. . .
```

```
    //test for adding wrong cases:
    //Check method in 'Club'(AddCourt, AddCoach and AddPlayer)
    //test case 1: add existing object in the club
    //test case 2: add wrong input parameter when adding court 'size' and 'type'
    //test case 3: add objects with reached to the max number
    //test result 1: 'input id already exists will be printed' for add existing object
    //test result 2: 'Ileagal type input!' or 'Ileagal size input!' will be printed for input court's wrong parameter
    //test result 3: 'Add failed! Court number reached max number' for add objects with max number
    void test03() {//****forgot print results
        cout << "test3: test for add member wrong case:" << endl;
        Club club(1, "Tennis Club", "Stary Miasto", 01);
        cout << "--------------------------------------------------------------------------------
" << endl;
        cout << "Wrong case 1: Trying to add an existing court, coach and player: " << endl;
        club.AddCourt(01,'I','L');
        //test for adding existing court
        club.AddCourt(01,'I','L');
        club.AddCoach("Jiaxu Luo", 01, 1);
        //test for adding existing coach
        club.AddCoach("Jiaxu Luo", 01, 1);
        club.AddPlayer("Haolin Li", 01, 1, 50.0);
        //test for adding existing player
        club.AddPlayer("Haolin Li", 01, 1, 50.0);
        //test for adding court in case of number exceed its max number
        cout << endl;

        cout << "--------------------------------------------------------------------------------
" << endl;
        cout << "Wrong case 2: Test for adding ileagal type and size parameter: " << endl;
        club.AddCourt(2, 'M', 'L');
        club.AddCourt(2, 'I', 'C');
        cout << endl;

        cout << "--------------------------------------------------------------------------------
" << endl;
        cout << "Wrong case 3: " << endl;
        cout << "Adding court when the club already existing " << MaxCourtNumber << " courts(max: " << MaxCourtNumber << "): " << endl;
        club.AddCourt(02,'I','L');
        club.AddCourt(03,'I','L');
        club.AddCourt(04,'I','L');
        club.AddCourt(05,'I','L');
        club.AddCourt(06,'O','M');//not successful
        //here I changed the max player number from 80 to 10 for easier testing
        cout << "Adding player when the player number reached to the max number:" << endl;
        club.AddPlayer("A", 2, 1, 1);
        club.AddPlayer("B", 3, 1, 1);
```

```
44.     club.AddPlayer("C", 4, 1, 1);
45.     club.AddPlayer("D", 5, 1, 1);
46.     club.AddPlayer("E", 6, 1, 1);
47.     club.AddPlayer("F", 7, 1, 1);
48.     club.AddPlayer("G", 8, 1, 1);
49.     club.AddPlayer("H", 9, 1, 1);
50.     club.AddPlayer("I", 10, 1, 1);
51.     club.AddPlayer("J", 11, 1, 1);//not successful***
52.     cout << "Adding coach when the coach number reached to the max number: " << endl;
53.     club.AddCoach("A", 2, 1);
54.     club.AddCoach("B", 3, 1);
55.     club.AddCoach("C", 4, 1);
56.     club.AddCoach("D", 5, 1);
57.     club.AddCoach("E", 6, 1);
58.     club.AddCoach("F", 7, 1);
59.     club.AddCoach("G", 8, 1);
60.     club.AddCoach("H", 9, 1);
61.     club.AddCoach("I", 10, 1);
62.     club.AddCoach("J", 11, 1);//not successful***
63.     club.printCourtList();
64.     club.printCoachList();
65.     club.printPlayerList();
66. }
```

```
test3: test for add member wrong case:
--------------------------------------------------------------------------------
Wrong case 1: Trying to add an existing court, coach and player:
current court number: 1
Input id already exists a court!
Current coach number: 1
Input id already exists a coach!
Current player number: 1
Check player exists
Input id already exists a player!

--------------------------------------------------------------------------------
Wrong case 2: Test for adding ileagal type and size parameter:
Ileagal type input!
Ileagal size input!

--------------------------------------------------------------------------------
Wrong case 3:
Adding court when the club already existing 5 courts(max: 5):
current court number: 2
current court number: 3
current court number: 4
current court number: 5
Add failed! Court number reached max number: 5!
Adding player when the player number reached to the max number:
Current player number: 2
Current player number: 3
Current player number: 4
Current player number: 5
Current player number: 6
Current player number: 7
Current player number: 8
Current player number: 9
Current player number: 10
Add failed! Player number reached max 10
Adding coach when the coach number reached to the max number:
Current coach number: 2
Current coach number: 3
Current coach number: 4
Current coach number: 5
Current coach number: 6
Current coach number: 7
Current coach number: 8
Current coach number: 9
Current coach number: 10
Add failed! Coach number reached max 10!
Court id: 1  type: Indoor  size: Large
```

44.

```
Add failed! Player number reached max 10
Adding coach when the coach number reached to the max number:
Current coach number: 2
Current coach number: 3
Current coach number: 4
Current coach number: 5
Current coach number: 6
Current coach number: 7
Current coach number: 8
Current coach number: 9
Current coach number: 10
Add failed! Coach number reached max 10!
Court id: 1  type: Indoor  size: Large
Court id: 2  type: Indoor  size: Large
Court id: 3  type: Indoor  size: Large
Court id: 4  type: Indoor  size: Large
Court id: 5  type: Indoor  size: Large
Coach name: Jiaxu Luo  id: 1  salary: 8511  experience: 1  qualification: D
Coach name: A  id: 2  salary: 8511  experience: 1  qualification: D
Coach name: B  id: 3  salary: 8511  experience: 1  qualification: D
Coach name: C  id: 4  salary: 8511  experience: 1  qualification: D
Coach name: D  id: 5  salary: 8511  experience: 1  qualification: D
Coach name: E  id: 6  salary: 8511  experience: 1  qualification: D
Coach name: F  id: 7  salary: 8511  experience: 1  qualification: D
Coach name: G  id: 8  salary: 8511  experience: 1  qualification: D
Coach name: H  id: 9  salary: 8511  experience: 1  qualification: D
Coach name: I  id: 10  salary: 8511  experience: 1  qualification: D
Player name: Haolin Li  id: 1  transfer fee: 68511  experience: 1  salary: 24511  win-rate: 50%  level: B
Player name: A  id: 2  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: B  id: 3  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: C  id: 4  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: D  id: 5  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: E  id: 6  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: F  id: 7  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: G  id: 8  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: H  id: 9  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
Player name: I  id: 10  transfer fee: 8511  experience: 1  salary: 2511  win-rate: 1%  level: D
```

```
//test method in 'Club'(getCourtNumber, getCoachNumber and getPlayerNumber)
//Add one court, one coach and one player in the club and check the current member number
//correct case: current member number in club will be printed
void test04() {//***
    cout << "Test 4: test method of getCourtNumber, getCoachNumber and getPlayerNumber: " << endl;

    Club club(1, "Tennis Club", "Zloty Tarsy", 01);
    club.AddCourt(01, 'I', 'L');
    club.AddCoach("Jiaxu Luo", 01, 1);
    club.AddPlayer("Haolin Li", 01, 1, 50.0);
    club.printCourtList();
    club.printCoachList();
    club.printPlayerList();
}
```

```
Test 4: test method of getCourtNumber, getCoachNumber and getPlayerNumber:
current court number: 1
Current coach number: 1
Current player number: 1
Court id: 1  type: Indoor  size: Large
Coach name: Jiaxu Luo  id: 1  salary: 9281  experience: 1  qualification: D
Player name: Haolin Li  id: 1  transfer fee: 69281  experience: 1  salary: 25281  win-rate: 50%  level: B
Player with id: 1 has been removed

Coach with id: 1 has been removed

Court with id: 1 has been removed

Club: Tennis Club has been removed
```

```
//test for update coach's and player's qualification and level while adding coach and player
//correct case:
//if input coach id is valid, the function will check this coach's experience, and upgrade his level and salary via experience
//if input player id is valid, the function will check this player's win-rate, and upgrade his level , transfer fee and salary via experience
//fail case:
//if input coach id is not vaild, an error message will be printed
```

```
54.  void test05() {
55.      cout << "test 5: update coach's and player's qualification and level: " << endl;
56.      Club club(1, "Tennis Club", "Zloty Tarsy", 01);
57.      club.AddCoach("Jiaxu Luo", 01, 1);
58.      club.AddPlayer("Haolin Li", 01, 1, 30);
59.      cout << "Coach's salary and level will be given automatically according to input experience:"
<< endl;
60.      club.printCoachList();
61.      cout << "Player's salary, transfer fee and level will be given automatically according to inpu
t win-rate:" << endl;
62.      club.printPlayerList();
63.  }
```

```
test 5: update coach's and player's qualification and level:
Current coach number: 1
Current player number: 1
Coach's salary and level will be given automatically according to input experience:
Coach name: Jiaxu Luo  id: 1  salary: 8504  experience: 1  qualification: D
Player's salary, transfer fee and level will be given automatically according to input win-rate:
Player name: Haolin Li  id: 1  transfer fee: 5504  experience: 1  salary: 2504  win-rate: 30%  level: D
请按任意键继续. . .
```

64.

```
65.  //test for print list of an empty club
66.  void test06() {
67.      cout << "test4: test for club empty object: " << endl;
68.      Club club(1, "Tennis Club", "Zloty Tarsy", 01);
69.      club.printCourtList();
70.      club.printCoachList();
71.      club.printPlayerList();
72.  }
```

```
test6: test for club empty object:
Court list is empty!
Coach list is empty!
Player list is empty!
请按任意键继续. . .
```

73.

```
74.  //test for checking print function in court's coach list and coach's player list
75.  //Special case: if the court's coach number is zero, an empty message will be printed
76.  //Correct case: if input id is valid, list information will be printed
77.  //Wrong case: input wrong id, an error message will be printed(Input id corresponds with no court
or coach)
78.  void test07() {
79.      cout << "test 7: Checking print function in court's coach list and coach's player list: " << e
ndl;
80.      Club club(1, "Tennis Club", "Zloty Tarsy", 01);
81.      //have assignment right now, test cases in the court's list
82.      club.AddCourt(01, 'O', 'S');
83.      club.AddCoach("Jiaxu Luo", 01, 1);
84.      club.AddCoach("Athas", 02, 2);
85.      club.AddPlayer("Haolin Li", 01, 1, 50.0);
86.      club.AddPlayer("Jenny", 02, 1, 70.0);
87.      //club.AssignCoachToCourt(01, 01);
88.      cout << "-------------------------------------------------------" << endl;
89.      cout << "Test special case: test for print current court's coach list with 0 coach: " << endl;
90.      club.PrintCoach(01);
91.      cout << "-------------------------------------------------------" << endl;
92.      cout << "Test special case: test for print current coach's player list with 0 player: " << end
l;
93.      club.PrintPlayer(01);
```

```
94.
95.         cout << "-----------------------------------------------------" << endl;
96.         cout << "Assignment: " << endl;
97.         club.AssignPlayerToCoach(01, 01);
98.         club.AssignCoachToCourt(01, 01);
99.         cout << "-----------------------------------------------------" << endl;
100.        cout << "Check coach list in club: " << endl;
101.        club.printCoachList();
102.
103.        cout << "-----------------------------------------------------" << endl;
104.        cout << "Test for correct case: " << endl;
105.        cout << "Check coach list in court: " << endl;
106.        club.PrintCoach(01);
107.
108.        cout << "-----------------------------------------------------" << endl;
109.        cout << "Check player list in club: " << endl;
110.        club.printPlayerList();
111.
112.        cout << "-----------------------------------------------------" << endl;
113.        cout << "Check player list in coach: " << endl;
114.        club.PrintPlayer(01);
115.        cout << "-----------------------------------------------------" << endl;
116.        cout << "Wrong case: " << endl;
117.        club.PrintCoach(02);
118.        club.PrintPlayer(03);
119. }
```

```
test 7: Checking print function in court's coach list and coach's player list:
current court number: 1
Current coach number: 1
Current coach number: 2
Current player number: 1
Current player number: 2
-----------------------------------------------------
Test special case: test for print current court's coach list with 0 coach:
The court's coach list is empty!
-----------------------------------------------------
Test special case: test for print current coach's player list with 0 player:
The coach's player list is empty!
-----------------------------------------------------
Assignment:
Check player exists
Player with id 1 has been assigned to coach with id 1 successfully!
Coach with id 1 has been assigned to court with id 1 successfully!
-----------------------------------------------------
Check coach list in club:
Coach name: Jiaxu Luo  id: 1  salary: 8984  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 10984  experience: 2  qualification: C
-----------------------------------------------------
Test for correct case:
Check coach list in court:
Coach name: Jiaxu Luo  id: 1  salary: 8984  experience: 1  qualification: D
-----------------------------------------------------
Check player list in club:
Player name: Haolin Li  id: 1  transfer fee: 50984  experience: 1  salary: 22984  win-rate: 50%  level: B
Player name: Jenny  id: 2  transfer fee: 120984  experience: 1  salary: 60984  win-rate: 70%  level: A
-----------------------------------------------------
Check player list in coach:
Player name: Haolin Li  id: 1  transfer fee: 50984  experience: 1  salary: 22984  win-rate: 50%  level: B
-----------------------------------------------------
Wrong case:
Input id corresponds with no court!
Input id corresponds with no coach!
请按任意键继续. . .
```

```
120.
121. //Test assignment
122. //Assign a coach to court, assign a player to coach
123. //Correct case: assigned successful information will be printed
124. //Wrong case 1: if input coach id or player id is not valid, an error message will be printed
125. //Wrong case 2: if the court's coach number or coach's player number reached to the max number. an
     error message will be printed
```

```
126. void test08() {
127.     cout << "Test 8: test for assignment, assign a coach to court, and assign a player to coach: "
     << endl;
128.     Club club(1, "Tennis Club", "Zloty Tarsy", 01);
129.     club.AddCourt(01, 'I', 'L');
130.
131.     club.AddCoach("Jiaxu Luo", 01, 1);
132.     club.AddCoach("A", 2, 1);
133.     club.AddCoach("B", 3, 1);
134.
135.     club.AddPlayer("Haolin Li", 01, 1, 50.0);
136.     club.AddPlayer("A", 2, 1, 1);
137.     club.AddPlayer("B", 3, 1, 1);
138.     club.AddPlayer("C", 4, 1, 1);
139.     club.AddPlayer("D", 5, 1, 1);
140.
141.     cout << "-----------------------------------------------------------" << endl;
142.     cout << "Correct assignment case: " << endl;
143.     club.AssignPlayerToCoach(01, 01);
144.     club.AssignCoachToCourt(01, 01);
145.     cout << "-----------------------------------------------------------" << endl;
146.     cout << "Wrong case 1: if input coach id or player id is not valid, an error message will be p
    rinted: " << endl;
147.     club.AssignPlayerToCoach(11, 11);
148.     cout << "-----------------------------------------------------------" << endl;
149.     cout << "Wrong case 2: the court's coach number or coach's player number reached to the max nu
    mber: " << endl;
150.     cout << "For court adding coach with max coach number: " << endl;
151.     club.AssignCoachToCourt(02, 01);
152.     club.AssignCoachToCourt(03, 01);
153.
154.     cout << "-----------------------------------------------------------" << endl;
155.     cout << "For coach adding player with max player number: " << endl;
156.     club.AssignPlayerToCoach(02, 01);
157.     club.AssignPlayerToCoach(03, 01);
158.     club.AssignPlayerToCoach(04, 01);
159.     club.AssignPlayerToCoach(05, 01);
160. }
```

```
Test 8: test for assignment, assign a coach to court, and assign a player to coach:
current court number: 1
Current coach number: 1
Current coach number: 2
Current coach number: 3
Current player number: 1
Current player number: 2
Current player number: 3
Current player number: 4
Current player number: 5
------------------------------------------------------------
Correct assignment case:
Check player exists
Player with id 1 has been assigned to coach with id 1 successfully!
Coach with id 1 has been assigned to court with id 1 successfully!
------------------------------------------------------------
Wrong case 1: if input coach id or player id is not valid, an error message will be printed:
Input id corresponds with no coach or player!
------------------------------------------------------------
Wrong case 2: the court's coach number or coach's player number reached to the max number:
For court adding coach with max coach number:
Coach with id 2 has been assigned to court with id 1 successfully!
This court has no remaining place for new coach!
------------------------------------------------------------
For coach adding player with max player number:
Check player exists
Player with id 2 has been assigned to coach with id 1 successfully!
Check player exists
Player with id 3 has been assigned to coach with id 1 successfully!
Check player exists
Player with id 4 has been assigned to coach with id 1 successfully!
Check player exists
This coach has no remaining place for new player!
请按任意键继续. . .
```

```
161.
162. //test for remove object from existing court, coach and player
163. //Correct case: removed successful information will be printed
164. //Wrong case: input wrong id(not existing member), an error message will be printed
165. void test09() {
166.     cout << "test5: remove a court, coach and player from existing object: " << endl;
167.     Club club(1, "Tennis Club", "Zloty Tarsy", 01);
168.     //no assignment right now, only test cases in the club's list
169.     club.AddCourt(01,'I','L');
170.     club.AddCourt(02,'I','L');
171.     club.AddCoach("Jiaxu Luo", 01, 1);
172.     club.AddCoach("Kindy", 02, 1);
173.     club.AddPlayer("Haolin Li", 01, 1, 50.0);
174.     club.AddPlayer("Tomasz Jay", 02, 1, 34.0);
175.     cout << "------------------------------------------------------" << endl;
176.     cout << "Before remove a court, coach and player: " << endl;
177.     club.printCourtList();
178.     club.printCoachList();
179.     club.printPlayerList();
180.
181.     cout << "------------------------------------------------------" << endl;
182.     cout << "After remove a court, coach and player: " << endl;
183.     club.RemoveCoach(01);
184.     club.RemoveCourt(01);
185.     club.RemovePlayer(01);
186.     club.printCourtList();
187.     club.printCoachList();
188.     club.printPlayerList();
189.     cout << "------------------------------------------------------" << endl;
190.     cout << "Wrong case(remove with wrong input id): " << endl;
```

```
    club.RemoveCourt(11);
    club.RemoveCoach(11);
    club.RemovePlayer(11);
}
```

```
test 9: remove a court, coach and player from existing object:
current court number: 1
current court number: 2
Current coach number: 1
Current coach number: 2
Current player number: 1
Current player number: 2
------------------------------------------------------------
Before remove a court, coach and player:
Court id: 1  type: Indoor  size: Large
Court id: 2  type: Indoor  size: Large
Coach name: Jiaxu Luo  id: 1  salary: 9225  experience: 1  qualification: D
Coach name: Kindy  id: 2  salary: 9225  experience: 1  qualification: D
Player name: Haolin Li  id: 1  transfer fee: 51225  experience: 1  salary: 23225  win-rate: 50%  level: B
Player name: Tomasz Jay  id: 2  transfer fee: 6225  experience: 1  salary: 3225  win-rate: 34%  level: D
------------------------------------------------------------
After remove a court, coach and player:
This coach already fired!
This court removed successful!
This player already fired!
Court id: 2  type: Indoor  size: Large
Coach name: Kindy  id: 2  salary: 9225  experience: 1  qualification: D
Player name: Tomasz Jay  id: 2  transfer fee: 6225  experience: 1  salary: 3225  win-rate: 34%  level: D
------------------------------------------------------------
Wrong case(remove with wrong input id):
Input court id is not exist!
Input coach id is not exists!
Input player id is not exists!
请按任意键继续. . .
```

```
//test for remove a court with existing coach, remove a coach with existing player
//Correct case: removed successful information will be printed
//Wrong case: if new input coach id is the removed coach id, an error message will be printed
void test10() {
    cout << "test 10: test for remove a court with existing coach, remove a coach with existing player: " << endl;
    Club club(1, "Tennis Club", "Zloty Tarsy", 01);
    //have assignment right now, test cases in the court's list
    club.AddCourt(01, 'I', 'L');
    club.AddCourt(02, 'I', 'L');

    club.AddCoach("Jiaxu Luo", 01, 1);
    club.AddCoach("Athas", 02, 1);

    club.AddPlayer("Haolin Li", 01, 1, 50.0);
    cout << "------------------------------------------------------------" << endl;
    cout << "Assignment: " << endl;
    club.AssignPlayerToCoach(01, 01);
    club.AssignCoachToCourt(01, 01);//with player 1
    club.AssignCoachToCourt(02, 01);//with no player
    cout << "------------------------------------------------------------" << endl;
    cout << "Before remove a court with existing coach: " << endl;
    club.printCoachList();
    club.printCourtList();
    cout << "------------------------------------------------------------" << endl;
    //club.printPlayerList();
    //cout << "Remove a coach with existing player: " << endl;
    //club.RemoveCoach(01);
    //cout << "After removement: " << endl;
    //club.printCoachList();
    //club.printPlayerList();
    cout << "After remove a court with existing coach: " << endl;
```

```
    club.RemoveCourt(01);
    club.printCourtList();
    club.printCoachList();
    cout << endl;
    cout << "-------------------------------------------------------" << endl;
    cout << "Before remove a coach with existing player: " << endl;
    club.printPlayerList();
    club.printCoachList();
    cout << "-------------------------------------------------------" << endl;
    cout << "After remove a coach with existing player: " << endl;
    club.RemoveCoach(01);
    club.printPlayerList();
    club.printCoachList();
}
//For Wrong case:
```

```
test 10: test for remove a court with existing coach, remove a coach with existing player:
current court number: 1
current court number: 2
Current coach number: 1
Current coach number: 2
Current player number: 1
-------------------------------------------------------

Assignment:
Check player exists
Player with id 1 has been assigned to coach with id 1 successfully!
Coach with id 1 has been assigned to court with id 1 successfully!
Coach with id 2 has been assigned to court with id 1 successfully!
-------------------------------------------------------
Before remove a court with existing coach:
Coach name: Jiaxu Luo  id: 1  salary: 9441  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9441  experience: 1  qualification: D
Court id: 1  type: Indoor  size: Large
Court id: 2  type: Indoor  size: Large
-------------------------------------------------------
After remove a court with existing coach:
Please find another court to place current court's coach(id:  1): 1
Lleagal operation! You can't assign this coach!
Court id: 1  type: Indoor  size: Large
Court id: 2  type: Indoor  size: Large
Coach name: Jiaxu Luo  id: 1  salary: 9441  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9441  experience: 1  qualification: D

-------------------------------------------------------
Before remove a coach with existing player:
Player name: Haolin Li  id: 1  transfer fee: 51441  experience: 1  salary: 23441  win-rate: 50%  level: B
Coach name: Jiaxu Luo  id: 1  salary: 9441  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9441  experience: 1  qualification: D
-------------------------------------------------------
After remove a coach with existing player:
Please enter another coach's id to place current coach's player: 1
Lleagal operation! You can't assign this player!
Player name: Haolin Li  id: 1  transfer fee: 51441  experience: 1  salary: 23441  win-rate: 50%  level: B
Coach name: Jiaxu Luo  id: 1  salary: 9441  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9441  experience: 1  qualification: D
请按任意键继续. . .
```

//For correct case:

```
test 10: test for remove a court with existing coach, remove a coach with existing player:
current court number: 1
current court number: 2
Current coach number: 1
Current coach number: 2
Current player number: 1
------------------------------------------------------------
Assignment:
Check player exists
Player with id 1 has been assigned to coach with id 1 successfully!
Coach with id 1 has been assigned to court with id 1 successfully!
Coach with id 2 has been assigned to court with id 1 successfully!
------------------------------------------------------------
Before remove a court with existing coach:
Coach name: Jiaxu Luo  id: 1  salary: 9797  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9797  experience: 1  qualification: D
Court id: 1  type: Indoor  size: Large
Court id: 2  type: Indoor  size: Large
------------------------------------------------------------
After remove a court with existing coach:
Please find another court to place current court's coach(id:  1): 2
Added all coaches with court id 2 successful!
Please find another court to place current court's coach(id:  2): 2
Added all coaches with court id 2 successful!
This court removed successful!
Court id: 2  type: Indoor  size: Large
Coach name: Jiaxu Luo  id: 1  salary: 9797  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9797  experience: 1  qualification: D

------------------------------------------------------------
Before remove a coach with existing player:
Player name: Haolin Li  id: 1  transfer fee: 51797  experience: 1  salary: 23797  win-rate: 50%  level: B
Coach name: Jiaxu Luo  id: 1  salary: 9797  experience: 1  qualification: D
Coach name: Athas  id: 2  salary: 9797  experience: 1  qualification: D
------------------------------------------------------------
After remove a coach with existing player:
Please enter another coach's id to place current coach's player: 2
Added all players with coach id 2 successful!
This coach already fired!
Player name: Haolin Li  id: 1  transfer fee: 51797  experience: 1  salary: 23797  win-rate: 50%  level: B
Coach name: Athas  id: 2  salary: 9797  experience: 1  qualification: D
请按任意键继续. . .
```

```
//test for reassignment
//Correct case:if input new id is valid, 'Added all players/coaches with coach/court id successful!
' will be printed
//Wrong case: if input new id is not valid, an error message will be printed
void test11() {
    cout << "Test 11: test for reassignment: " << endl;
    Club club(1, "Tennis Club", "Zloty Tarsy", 01);
    club.AddCourt(01, 'I', 'L');
    club.AddCoach("Jiaxu Luo", 01, 1);
    club.AddCoach("A", 2, 1);
    club.AddPlayer("Haolin Li", 01, 1, 50.0);
    club.AddPlayer("A", 02, 1, 50.0);
    cout << "------------------------------------------------------------" << endl;
    cout << "Assignment: " << endl;
    club.AssignPlayerToCoach(01, 01);
    club.AssignCoachToCourt(01, 01);//with player 1
    cout << "------------------------------------------------------------" << endl;
    cout << "Correct case: " << endl;
    club.ReassignCoachToCourt(02, 01);
    club.ReassignPlayerToCoach(02, 01);
    cout << "------------------------------------------------------------" << endl;
    cout << "Wrong case: " << endl;
    club.ReassignCoachToCourt(02, 02);
    club.ReassignPlayerToCoach(02, 03);
```

268. }

```
Test 11: test for reassignment:
current court number: 1
Current coach number: 1
Current coach number: 2
Current player number: 1
Current player number: 2
------------------------------------------------------------

Assignment:
Check player exists
Player with id 1 has been assigned to coach with id 1 successfully!
Coach with id 1 has been assigned to court with id 1 successfully!
------------------------------------------------------------

Correct case:
Added all coaches with court id 1 successful!
Added all players with coach id 1 successful!
------------------------------------------------------------

Wrong case:
INPUT COURT ID NOT EXIST!
INPUT COACH ID NOT EXISTS!
请按任意键继续. . .
```

```
1.  //add dynamic club and test destructor
2.  void test12() {
3.      cout << "test 12: test for destructor(automatic): " << endl;
4.      Club club1(1, "Tennis Club", "Zloty Tarsy", 01);
5.      club1.AddCourt(01, 'I', 'L');
6.      club1.AddCoach("Jiaxu Luo", 01, 1);
7.      club1.AddPlayer("Haolin Li", 01, 1, 50.0);
8.      cout << "------------------------------------------------------------" << endl;
9.      cout << "test for destructor(dynamic): " << endl;
10.     Club* club2 = new Club(2, "Tennis CLUB", "Stary Miasto", 02);
11.     club2->AddCourt(02, 'I', 'L');
12.     club2->AddCoach("Luo Jiaxu", 02, 1);
13.     club2->AddPlayer("Li Haolin", 02, 1, 50.0);
14.     cout << "------------------------------------------------------------" << endl;
15.     delete club2;
16. }
```

```
test 12: test for destructor(automatic):
current court number: 1
Current coach number: 1
Current player number: 1
------------------------------------------------------------

test for destructor(dynamic):
current court number: 1
Current coach number: 1
Current player number: 1
------------------------------------------------------------
Player with id: 2 has been removed

Coach with id: 2 has been removed

Court with id: 2 has been removed

Club: Tennis CLUB has been removed

Player with id: 1 has been removed

Coach with id: 1 has been removed

Court with id: 1 has been removed

Club: Tennis Club has been removed

请按任意键继续. . .
```

269.

```
//remove a coach and check both coachlist in club and court
void test13() {
    cout << "test 13: remove a coach which had been assign to a court, check coach list in club and court: " << endl;
    Club club(1, "Tennis Club", "Zloty Tarsy", 01);
    //no assignment right now, only test cases in the club's list
    club.AddCourt(01, 'I', 'L');
    club.AddCourt(02, 'I', 'L');
    club.AddCoach("Jiaxu Luo", 01, 1);
    club.AddCoach("Kindy", 02, 1);
    //club.AddPlayer("Haolin Li", 01, 1, 50.0);
    //club.AddPlayer("Tomasz Jay", 02, 1, 34.0);
    cout << "------------------------------------------------------------" << endl;
    cout << "assignment: " << endl;
    club.AssignCoachToCourt(01, 01);
    cout << "------------------------------------------------------------" << endl;
    cout << "Before remove coach: " << endl;
    cout << "list in club: " << endl;
    //club.printCourtList();
    club.printCoachList();
    cout << "list in court with id 1: " << endl;
    club.PrintCoach(01);
    cout << "------------------------------------------------------------" << endl;
    cout << "After remove a coach: " << endl;
    cout << "list in club: " << endl;
    club.RemoveCoach(01);
    //club.RemoveCourt(01);
    //club.RemovePlayer(01);
```

```
        //club.printCourtList();
        club.printCoachList();
        cout << "list in court with id 1: " << endl;
        club.PrintCoach(01);
        //club.printPlayerList();
    }
```

```
test 13: remove a coach which had been assign to a court, check coach list in club and court:
current court number: 1
current court number: 2
Current coach number: 1
Current coach number: 2
------------------------------------------------------

assignment:
Coach with id 1 has been assigned to court with id 1 successfully!
------------------------------------------------------

Before remove coach:
list in club:
Coach name: Jiaxu Luo  id: 1  salary: 9996  experience: 1  qualification: D
Coach name: Kindy  id: 2  salary: 9996  experience: 1  qualification: D
list in court with id 1:
Coach name: Jiaxu Luo  id: 1  salary: 9996  experience: 1  qualification: D
------------------------------------------------------

After remove a coach:
list in club:
This coach already fired!
Coach name: Kindy  id: 2  salary: 9996  experience: 1  qualification: D
list in court with id 1:
The court's coach list is empty!
Coach with id: 2 has been removed

Court with id: 1 has been removed

Court with id: 2 has been removed

Club: Tennis Club has been removed
```

```
    //remove a player and check both playerlist in club and coach
    void test14() {
        cout << "test 14: remove a player which had been assign to a coach, check player list in club and court: " << endl;
        Club club(1, "Tennis Club", "Zloty Tarsy", 01);
        //no assignment right now, only test cases in the club's list
        //club.AddCourt(01, 'I', 'L');
        //club.AddCourt(02, 'I', 'L');
        club.AddCoach("Jiaxu Luo", 01, 1);
        //club.AddCoach("Kindy", 02, 1);
        club.AddPlayer("Haolin Li", 01, 1, 50.0);
        club.AddPlayer("Tomasz Jay", 02, 1, 34.0);
        cout << "------------------------------------------------------" << endl;
        cout << "assignment: " << endl;
        club.AssignPlayerToCoach(01, 01);
        cout << "------------------------------------------------------" << endl;
        cout << "Before remove player: " << endl;
        cout << "list in club: " << endl;
        //club.printCourtList();
        club.printPlayerList();
        cout << "list in coach with id 1: " << endl;
        club.PrintPlayer(01);
        cout << "------------------------------------------------------" << endl;
        cout << "After remove a player: " << endl;
        cout << "list in coach: " << endl;
        club.RemovePlayer(01);
        //club.RemoveCourt(01);
        //club.RemovePlayer(01);
```

```
28.     //club.printCourtList();
29.     club.printPlayerList();
30.     cout << "list in court with id 1: " << endl;
31.     club.PrintPlayer(01);
32.     //club.printPlayerList();
33. }
```

```
C:\Users\alf13\source\repos\real_project\x64\Debug\real_project.exe
test 14: remove a player which had been assign to a coach, check player list in club and court:
Current coach number: 1
Current player number: 1
Current player number: 2
------------------------------------------------------------

assignment:
Check player exists
Player with id 1 has been assigned to coach with id 1 successfully!
------------------------------------------------------------

Before remove player:
list in club:
Player name: Haolin Li  id: 1  transfer fee: 66610  experience: 1  salary: 22610  win-rate: 50%  level: B
Player name: Tomasz Jay  id: 2  transfer fee: 6610  experience: 1  salary: 2610  win-rate: 34%  level: D
list in coach with id 1:
Player name: Haolin Li  id: 1  transfer fee: 66610  experience: 1  salary: 22610  win-rate: 50%  level: B
------------------------------------------------------------

After remove a player:
list in coach:
This player already fired!
Player name: Tomasz Jay  id: 2  transfer fee: 6610  experience: 1  salary: 2610  win-rate: 34%  level: D
list in court with id 1:
The coach's player list is empty!
Player with id: 2 has been removed

Coach with id: 1 has been removed

Club: Tennis Club has been removed

请按任意键继续. . .
```

35.