

Exercițiu

Să se implementeze două clase `Jucator` și `Echipa`.

Clasa `Jucator` reprezintă persoane ce practică un sport de echipă, identificate printr-un nume și un număr de pe tricou. Clasa va avea următoarea structură:

- două atribute private, `nume` de tip pointer la `char` și `nrTricou` de tipul `int`, și care au valorile implicite “Unknown” și respectiv -1.
- constructor fără parametri, constructor cu parametri, constructor de copiere, metode *getter* pentru atributele private.
- operatorul de atribuire supraîncărcat.
- o metodă `citire()` ce realizează citirea datelor de identificare ale unui jucător dintr-un flux de intrare.
- o metodă `afisare()` ce afișează conținutul (datele de identificare) ale unui jucător.

Fișierul `jucator.h` ar putea conține:

```
#ifndef __JUCATOR_H_
#define __JUCATOR_H_

#include <iostream>
#include <string.h>

using namespace std;

#define MAX_NUME 15 // numarul maxim de caractere al numelui

class Jucator {
    char* nume; /* numele unui jucator */
    int nrTricou; /* numarul de pe tricou */

public:
    Jucator();
    Jucator(const char*, int);
    Jucator(const Jucator&);

    ~Jucator();

    Jucator& operator= (const Jucator&);

    char* getNume() const;
    int getNrTricou() const;

    void citire(istream& in);
    void afisare(ostream& out);
};

#endif // __JUCATOR_H_
```

Clasa `Echipa` reprezintă o mulțime compusă dintr-un număr de jucători ce practică un sport de echipă. Clasa va conține următoarele elemente:

- constructor cu parametri, constructor de copiere.
- destructor.
- operatorul de atribuire supraîncărcat.
- o metodă `setJucator(int, const Jucator&)` ce înlocuiește jucătorul aflat în cadrul echipei pe o anumită poziție specificată de primul parametru al metodei cu jucătorul specificat de cel de-al doilea parametru.
- o metodă `getJucator(int)` ce returnează o referință către jucătorul ce se află pe o anumită poziție în cadrul echipei, indicată de către parametrul metodei.
- o metodă `getTricou(int)` ce returnează un pointer către obiectul jucător ce poartă tricoul cu numărul indicat de către parametrul metodei.
- o metodă `citire()` ce citește un număr specificat de membrii/jucători din care se compune echipa.
- o metodă `afisare()` ce afișează componența echipei.

Fișierul `echipa.h` ar putea conține:

```
#ifndef __ECHIPA_H_
#define __ECHIPA_H_

#include "jucator.h"

class Echipa {
    char nume[MAX_NUME]; /* numele echipei */
    Jucator** membri;    /* un tablou de pointeri la jucatori */
    int nrMembri;        /* numarul de membri ai echipei */

public:
    Echipa(const char*, int);
    Echipa(const Echipa&);

    ~Echipa();

    Echipa& operator= (const Echipa&);

    void setJucator(int, const Jucator&);
    Jucator& getJucator(int) const;
    Jucator* getTricou(int) const;

    void citire(istream&);
    void afisare(ostream&);
};

#endif // __ECHIPA_H_
```

Observație. Proiectul va conține 5 fișiere: `Jucator.h`, `Jucator.cpp`, `Echipa.h`, `Echipa.cpp` și `test-echipa.cpp`.

Referințe:

Conținutul fișierului `test-echipa.cpp` ar putea fi:

```
#include "echipa.h"

int main() {
    Echipa team("Avantul", 4);
    Jucator juc;
    int p;

    team.citire(cin);
    team.afisare(cout);
    cout << " ++++++ \n";

    cout << "Schimbare jucator: ";
    cin >> p;
    juc.citire(cin);

    team.setJucator(p, juc);
    team.afisare(cout);

    return 0;
}
```

Să se modifice metodele `citire()` și `afisare()` din clasele `Jucator` și `Echipa`, astfel încât funcționalitatea acestora să fie preluată de către operatorii de “<<” (inserție în stream) și “>>” (extracție din stream) supraîncărcați pentru ambele clase.