Универзитет „Св. Кирил и Методиј“ - Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Object oriented programming

Exercises 4

Version 1.0, 3 March, 2017

Table of Contents

1. Composition

1.1. Date

Define a class for `Mate` that is described with name, age and gender.

Then, define a class `Date` that keeps data for two mates.

In this class, implement a function `bool isSuccess()` that should compute the success of the date. The date is successful if the sum of sums of the ASCII values of the names of the mates is even number. Also implement a function that will print the mates of the date in format:

```
Date between:
[Name] [Age] [Gender]
[Name] [Age] [Gender]
```

Solution `oop_av41_en.cpp`

```
#include <iostream>
#include <cstring>
using namespace std;

enum gender {
    male,
    female,
    other
};

class Mate {
private:
    char name[100];
    int age;
    gender gen;
public:
    Mate() {}

    Mate(const char *n, int a, gender g) {
        strcpy(name, n);
        age = a;
        gen = g;
    }

    int getNameNumber() {
        int sum = 0;
        char *n = name;
        while(*n) {
            sum += *n;
            ++n;
        }
        return sum;
    }


};

class Date {
private:
    Mate m1;
    Mate m2;
public:

    Date(Mate& _m1, Mate& _m2) {
        m1 = _m1;
        m2 = _m2;
    }

    bool isSuccess() {
        return (m1.getNameNumber() + m2.getNameNumber()) % 2 == 0;
    }
};


int main() {
    Mate m1("Barby", 20, female);
    Mate m2("Ken", 25, male);
    Mate m3("Sam", 23, other);

    Date date(m1, m2);
    if(date.isSuccess()) {
        cout << "The date was success, lets make another one..." << endl;
    } else {
        cout << "This did not work out, good bye!" << endl;
    }

    return 0;

}
```

1.2. Team

Define a class `Team` that is described with name, year of founding and the city. Define a class `Game` that keeps information for the host and the guest (objects of class `Team`), goals scored by the host and goals scored by the guest.

Define a function `rematch` that as arguments accepts two objects of class `Game` and checks if one of them is a rematch of the other. A match is a rematch if the host of the first game is a guest in the second game, and vice versa. If the match is a rematch, return the aggregate winner of the fixture (the one that scored more goals on aggregate).

Решение `oop_av42_en.cpp`

```
#include <iostream>
#include <cstring>
using namespace std;

class Team {
private:
    char name[100];
    int yearFounded;
    char city[50];
public:
    Team() {
    }
    Team(const char* n, int yf, const char*c) {
        strcpy(name, n);
        yearFounded = yf;
        strcpy(city, c);
    }

    bool equalTeam(Team& team) {
        return strcmp(team.name, name) == 0;
    }

    void print() {
        cout << name << " " << yearFounded << " " << city << endl;
    }
};

class Game {
private:
    Team host;
    Team guest;
    int goalsHost;
    int goalsGuest;
public:
    Game(Team& host, Team& guest, int gh, int gg) {
        this->host = host;
        this->guest = guest;
        goalsHost = gh;
        goalsGuest = gg;
    }

    bool equalsGuest(Game& game) {
        return host.equalTeam(game.guest);
    }

    bool equalHost(Game& game) {
        return guest.equalTeam(game.host);
    }
```

```
    Team winner(Game game) {
        int goals1 = goalsHost + game.goalsGuest;
        int goals2 = goalsGuest + game.goalsHost;
        if(goals1 > goals2) {
            return host;
        } else {
            return guest;
        }
    }
};

Team winner(Game& game1, Game& game2) {
    if(game1.equalHost(game2) && game1.equalsGuest(game2)) {
        return game1.winner(game2);
    } else {
        return Team("Dummy", 0, "");
    }
}

int main() {
    Team t1("Barcelona", 1900, "Barcelona");
    Team t2("Real Madrid", 1890, "Madrid");
    Team t3("Elche", 1950, "Elche");

    Game g1(t1, t2, 5, 0);
    Game g2(t2, t1, 2, 2);
    Game g3(t1, t3, 6, 2);

    Team w = winner(g1, g3);
    w.print();
    return 0;
}
```

2. Source code of the examples and problems

https://github.com/finki-mk/SP/

Source code ZIP