

Lista de Exercícios 6 (2023/2)

Este trabalho consiste em resolver a lista de exercícios das páginas a seguir, em C++.

Para a resolução e entrega devem ser seguidas as seguintes regras:

- criar uma pasta com o nome do aluno no formato *camelHump* (por exemplo, para João Pedro da Silva, usar `JoaoPedroDaSilva`);
- dentro dessa pasta criar programas em C++ para resolver cada um dos exercícios, salvando o código-fonte em um arquivo com o nome `Exercicio` seguido do número do exercício com três dígitos (por exemplo, `Exercicio001.cpp`, `Exercicio002.cpp`, ..., `Exercicio100.cpp`);
- no início de cada arquivo em C++, incluir um comentário informando o nome do arquivo, o nome do autor, a finalidade do programa e a versão (ou data) de criação (ou atualização);
- se houver dados para serem lidos, eles devem ser lidos na mesma ordem em que eles são citados no enunciado, escolhendo os tipos numéricos adequadamente;
- escrever os resultados sempre na mesma ordem em que eles são citados no enunciado, escolhendo os tipos numéricos adequadamente (números reais devem ser apresentados sempre com 4 casas decimais, salvo se indicado de outra forma);
- na versão final, tomar o cuidado de não imprimir nada diferente da saída esperada (não devem aparecer, por exemplo, mensagens pedindo que o usuário forneça ou digite determinado valor no terminal);
- a entrega deverá ser feita no dia e horário informado pelo professor em sala de aula e/ou definida na opção de entrega da plataforma moodle da PUCRS;
- cada aluno deverá submeter os códigos-fontes compactados no formato ZIP, usando o mesmo nome da pasta (por exemplo, para João Pedro da Silva, o arquivo compactado deverá chamar-se `JoaoPedroDaSilva.zip`).

3. Complete o código a seguir em C++, implementando os métodos da classe `Campeonato` de forma externa à definição da classe, usando como estrutura principal uma **lista duplamente encadeada** de nodos que contém referências para objetos da classe `Equipe`. A inserção na lista deve ser feita de forma ordenada, do maior (início da lista) para o menor número de pontos (final da lista). Considerando que NÃO há mais informações sobre o desempenho das equipes, em caso de empate entre duas ou mais equipes em número de pontos, as equipes devem aparecer na lista por ordem alfabética crescente. Para a implementação da classe `Campeonato`, a classe `Equipe` (cuja implementação é apresentada a seguir) deve ser usada como nodo da lista. Neste código, há uma função `main` que executa um teste e que deve ser mantida sem alterações.

Os métodos que não foram fornecidos e que devem ser implementados na classe `Campeonato` são:

- `Nodo(Equipe *e)`: cria um nodo atribuindo e ao campo `equipe` desse nodo e “zerando” encadeamentos;
- construtor: cria uma lista vazia de equipes;
- destrutor: desaloca os nodos da estrutura, simplesmente chamando `limpa()`;
- `obtemNumEquipes()`: retorna o número de equipes na estrutura de dados;
- `obtemEquipe(int indice)`: retorna o endereço da equipe que está no nodo na posição índice da lista ou `nullptr` se o índice for inválido;
- `insere(string c, int p)`: cria uma equipe (formada por nome do clube e número de pontos) e a insere na lista, simplesmente chamando `insere(new Equipe(c,p))`;
- `insere(Equipe *e)`: recebe o endereço de uma equipe e insere-a na lista de clubes do campeonato em ordem decrescente de número de pontos (em caso de igualdade no número de pontos o critério seguinte é a ordem alfabética crescente);
- `limpa()`: desaloca cada um dos nodos de equipes da estrutura, bem como a equipe do nodo, deixando a lista vazia.

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

class Equipe {
private:
    string clube;
    int pontos;
public:
    Equipe(string c="", int p=0) {
        clube = c;
        pontos = p;
    }
    string obtemClube() { return clube; }
    void defineClube(string c) { clube = c; }
    int obtemPontos() { return pontos; }
    void definePontos(int p) { pontos = p; }
    string toString() {
        stringstream ss;
        ss << clube << ":\n" << pontos << "\npontos";
        return ss.str();
    }
    string toCSV() {
        stringstream ss;
        ss << clube << ";" << pontos;
        return ss.str();
    }
};

class Campeonato {
private:
    class Nodo {
    public:
        Equipe *equipe;
        Nodo *prox, *ant;
        Nodo(Equipe *e);
    };
    Nodo *primeiro;
    int numEquipes;
public:
    Campeonato();
    ~Campeonato();
    int obtemNumEquipes();
    Equipe *obtemEquipe(int indice);
    void insere(string c, int p);
    void insere(Equipe *e);
    void limpa();
};
```

```

// --- SOLUCAO (inicio) ---
// ...
// --- SOLUCAO (fim) ---

void mostraCampeonato(Campeonato *c) {
    int tam = c->obtemNumEquipes();
    for (int i=0; i<tam; ++i) {
        Equipe *e = c->obtemEquipe(i);
        cout << e->toString() << endl;
    }
    cout << "-----" << endl;
}

int main() {
    Campeonato *brasileirao2021 = new Campeonato();
    mostraCampeonato(brasileirao2021);
    Equipe *e = new Equipe("Chapecoense",12);
    brasileiro2021->insere(e);
    mostraCampeonato(brasileirao2021);
    brasileiro2021->insere("Santos",28);
    brasileiro2021->insere("Bragantino",41);
    brasileiro2021->insere("Atletico_Goianiense",31);
    mostraCampeonato(brasileirao2021);
    brasileiro2021->insere("Flamengo",42);
    brasileiro2021->insere("Atletico-MG",53);
    brasileiro2021->insere("Internacional",36);
    brasileiro2021->insere("Fluminense",33);
    brasileiro2021->insere("Juventude",28);
    brasileiro2021->insere("Sport",26);
    brasileiro2021->insere("Cuiaba",31);
    brasileiro2021->insere("Gremio",23);
    mostraCampeonato(brasileirao2021);
    brasileiro2021->insere("Corinthians",37);
    brasileiro2021->insere("Athletico-PR",33);
    brasileiro2021->insere("Palmeiras",40);
    brasileiro2021->insere("Sao_Paulo",30);
    brasileiro2021->insere("Ceara",29);
    brasileiro2021->insere("Fortaleza_EC",39);
    brasileiro2021->insere("Bahia",27);
    brasileiro2021->insere("America-MG",31);
    mostraCampeonato(brasileirao2021);
    delete brasileiro2021;
    return 0;
}

```

Exemplo:

Entrada	Saída
	<pre> ----- Chapecoense: 12 pontos ----- Bragantino: 41 pontos Atletico Goianiense: 31 pontos Santos: 28 pontos Chapecoense: 12 pontos ----- Atletico-MG: 53 pontos Flamengo: 42 pontos Bragantino: 41 pontos Internacional: 36 pontos Fluminense: 33 pontos Atletico Goianiense: 31 pontos Cuiaba: 31 pontos Juventude: 28 pontos Santos: 28 pontos Sport: 26 pontos Gremio: 23 pontos Chapecoense: 12 pontos ----- Atletico-MG: 53 pontos Flamengo: 42 pontos Bragantino: 41 pontos Palmeiras: 40 pontos Fortaleza EC: 39 pontos Corinthians: 37 pontos Internacional: 36 pontos Athletico-PR: 33 pontos Fluminense: 33 pontos America-MG: 31 pontos Atletico Goianiense: 31 pontos Cuiaba: 31 pontos Sao Paulo: 30 pontos Ceara: 29 pontos Juventude: 28 pontos Santos: 28 pontos Bahia: 27 pontos Sport: 26 pontos Gremio: 23 pontos Chapecoense: 12 pontos ----- </pre>