

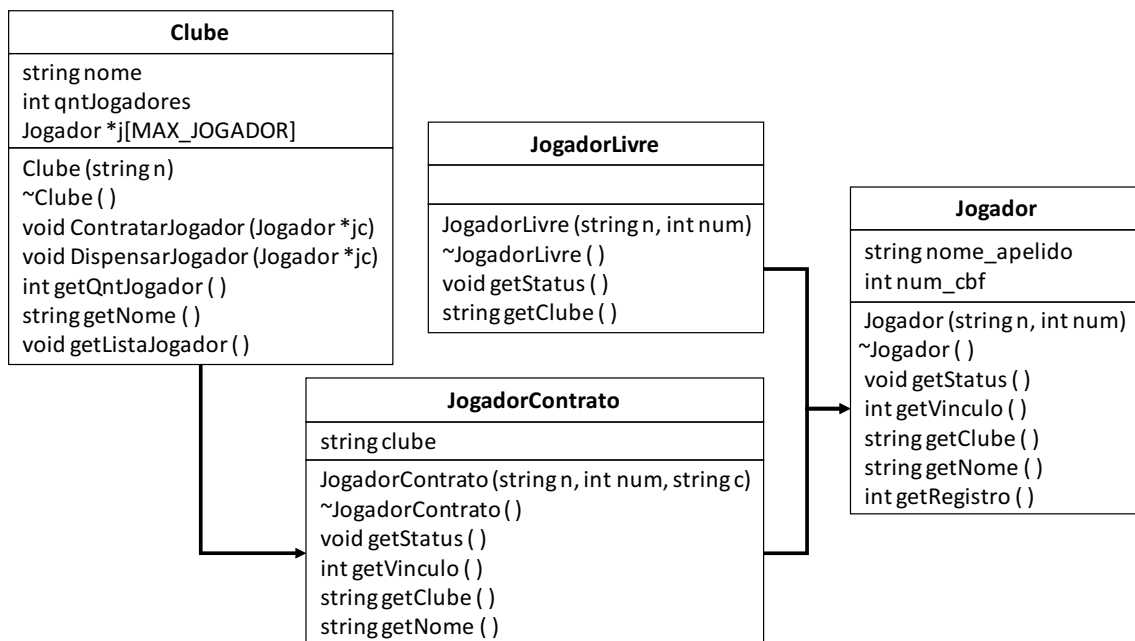
Trabalho Prático 3

1. Introdução

O objetivo deste trabalho é praticar a programação utilizando a linguagem C++ através da implementação de um sistema de registro de jogadores, que é baseado na relação de atletas participantes do campeonato gaúcho de 2016. O sistema deve ser implementado utilizando exclusivamente a linguagem C++ e exercitando os conceitos estudados em aula, com ênfase em classes, herança e polimorfismo.

2. Funcionamento e detalhes técnicos

As classes que deverão ser implementadas, junto com seus vínculos são mostradas abaixo:



O sistema deve permitir ao operador realizar as funcionalidades apresentadas a seguir no menu:

MENU

- 1 - Escolha o clube
 - 1.1 - Mostrar lista de jogadores
 - 1.2 - Mostrar detalhes de um jogador
 - 1.3 - Mostrar quantidade de jogadores
 - 1.4 - Contratar um jogador

- 1.5 - Dispensar um jogador
 - 1.6 - Voltar ao menu principal
 - 2 - Mostrar lista dos clubes
 - 3 - Mostrar lista de jogadores sem clube
 - 4 - Sair do programa
-

Onde a descrição de cada opção é:

1 - O operador deve escolher um clube, onde será realizada determinadas operações (entre **1.1** e **1.6**).

1.1 - Esta opção deve mostrar na tela uma lista com os nomes/apelidos dos jogadores registrados pelo clube selecionado.

1.2 - Esta opção deve mostrar na tela os detalhes de um jogador selecionado a partir de seu nome (se este for do clube selecionado), onde aparecerá nome ou apelido, o número de registro da CBF e o nome do clube em que este jogador tem vínculo.

1.3 - Esta opção deve mostrar na tela a quantidade de jogadores que o clube selecionado tem registrado. Como a base de dados reflete aos registrados no Gauchão de 2016, o número máximo de jogadores por clube deve ser 32 (MAX_JOGADOR = 32).

1.4 - Nesta opção, o clube selecionado pode contratar um jogador (selecionado pelo número de registro da CBF), se este não estiver vínculo com outro clube (i.e., *JogadorLivre*) e o clube tiver menos de 32 registrados, uma vez que esse é o máximo de jogadores que o campeonato gaúcho aceita registrar. Observação: Se um jogador puder ser contratado, ele deverá trocar seu vínculo de *JogadorLivre* para *JogadorContratado* para ser vinculado a um Clube.

1.5 - Nesta opção, o clube selecionado pode dispensar um jogador (selecionado pelo número de registro da CBF). Lembre-se que ele deverá trocar seu vínculo de *JogadorContratado* para *JogadorLivre*.

1.6 - Esta opção não é obrigatória. Ela somente existirá se o programador separar o menu em duas partes (principal e secundária), caso o programador escolha mostrar todas as opções de uma só vez ao usuário, essa opção não existe.

2 - Esta opção deve mostrar na tela a lista de clubes registrados no gauchão 2016.

3 - Esta opção deve mostrar na tela os detalhes de um jogador sem clube, que são o número de registro na CBF e o nome do atleta.

4 - O programa deve acabar se esta opção for selecionada. Lembre-se de atualizar o arquivo "BD.txt" antes de finalizar o programa.

Outros detalhes importantes são:

- No arquivo "BD.txt", o padrão que segue é o seguinte:
 - ✓ Caso o primeiro caractere seja um número, isso significa que o padrão é "NUM_CBF NOME_APELIDO", onde o NUM_CBF é um número de registro com 6 caracteres e o NOME_APELIDO contém uma string com o nome ou apelido que está registrado na federação gaúcha, logo devemos considerar a(s) string(s) até o final da linha.

- ✓ Caso o primeiro caractere seja uma letra, isso significa que o padrão é “NOME_CLUBE”, onde este é uma string que contém o nome do clube.
- Devemos inicializar o programa com os dados lidos do arquivo “BD.txt”, e ao final, devemos atualizar o mesmo arquivo antes de sair do programa.
- Jogadores vinculados ao Sindicato dos atletas profissionais no Estado do Rio Grande do Sul (SIAPERGS) são os jogadores sem contrato (*JogadorLivre*) que pode ser contratado por algum clube.
- É vetado a contratação de jogadores que possuem vínculos com outros clubes (*JogadorContratado*).
- Na função principal (main) deveremos trabalhar **somente** com os ponteiros para as classes definidas abaixo:
 - ✓ Jogador *j[qntJ];
 - O vetor de ponteiros deve ser definido por **qntJ**, que indica *a quantidade exata de jogadores* lidos do arquivo “BD.txt”.
 - ✓ Clube *c[qntC];
 - O vetor de ponteiros deve ser definido por **qntC**, que indica *a quantidade exata de clubes jogadores* lidos do arquivo “BD.txt”.
 - **Não** devemos considerar o SIAPERGS.
- Utilizar **POLIMORFISMO** para os objetos do tipo *Jogador* na função main se comportar de acordo com o seu tipo, *JogadorLivre* ou *JogadorContratado*.
- Lembre-se de criar um programa que seja **o mais eficiente possível** e **não utilizar memória desnecessária na criação dos ponteiros**. Além disso, na apresentação do programa será trocado o arquivo de leitura, o qual terá o mesmo padrão e nome (“BD.txt”), mas terá uma quantidade de clubes e jogadores diferentes, aos quais o seu programa deve se adaptar a esses novos tamanhos de maneira eficiente.

3. Avaliação

Leia com atenção os critérios de avaliação:

- Implementação adequada da lógica.
- Leitura apropriada das entradas do usuário e apresentação das informações na forma de texto no terminal.
- O trabalho será em **duplas** ou **individual**. Os arquivos contendo os códigos-fontes (.cpp) e um Makefile devem ser compactados e submetidos pelo Moodle até a data e hora especificadas. O arquivo compactado deve ter os nomes e os últimos sobrenomes de ambos os alunos, da seguinte forma:
 - ✓ **nome1_sobrenome1_nome2_sobrenome2.zip**
- O código-fonte deve estar identado e comentado adequadamente.
- A nota do trabalho depende da apresentação deste no laboratório, na data marcada. Trabalhos entregues, mas não apresentados terão sua nota anulada automaticamente.

Durante a apresentação será avaliado o domínio da resolução do problema, podendo inclusive ser possível invalidar o trabalho quando constatada a falta de conhecimento sobre o código implementado.

- A avaliação será dividida da seguinte forma:
 - ✓ Criação de um Makefile (5%);
 - ✓ Criação do menu (5%);
 - ✓ Criação das classes (15%);
 - ✓ Leitura e atualização do arquivo "BD.txt" de acordo com o histórico do programa (15%);
 - ✓ Inicialização mínima do vetor de ponteiros de Clubes e Jogadores (15%);
 - ✓ Utilização correta de polimorfismo (10%);
 - ✓ Programa totalmente funcional (15%);
 - ✓ Apresentação em sala de aula (20%).
- **A cópia parcial ou completa do trabalho terá como consequência a atribuição de nota ZERO ao trabalho dos alunos envolvidos.**