BCC202 – Estruturas de Dados I (2015-02)

Departamento de Computação - Universidade Federal de Ouro Preto - MG

Aula Prática 02 – TADs e Alocação Dinâmica

- **Data de entrega: 15/11/2015 até 23:55. O que vale é o horário do Moodle, e não do *seu*, ou do *meu*, relógio!!!**

- Procedimento para a entrega:.

1. Crie uma pasta para a questão: **PrimeiroNome-UltimoNome-Questao** (exemplo: `reinaldo-fortes-1`).
2. Crie os arquivos de codigo fonte necessarios para a solução da questão (arquivos **`.c`** e **`.h`**)
3. Compile na linha de comando usando **`g++ *.cpp -o prog.exe`** .
4. Execute usando redirecionamento de entrada: **`./prog.exe < entrada.txt`**.
5. Apague os arquivos gerados na compilação e o arquivo de entrada (**mantenha apenas os arquivos `.cpp` e `.h`**).
6. Compacte a pasta criada no item 1 (só serão aceitas arquivos compactados com extensão *.zip*).
7. Faça a entrega do arquivo compactado no moodle, na tarefa destinada à prática e questão correspondentes.

- A cada etapa, verifique se o resultado está conforme o esperado.
- **Não utilize caracteres acentuados ou especiais para nomes de pastas e arquivos**.

- Bom trabalho!

Questão 01

Implemente um TAD **`TJogador`**. Cada jogador possui os campos: ***Nome*** (no maximo 50 caracteres, sem espaços e acentos), ***Jogos*** (número de jogos em que atuou), ***Pontos*** (número de pontos marcados), ***Faltas*** (número de faltas marcadas) e ***Cartoes*** (numero de cartões recebidos). Este TAD deverá possuir as seguintes operações:

- **create*TJogador***: função que aloca dinamicamente uma variável do tipo TJogador.
- **get*Campo***: função que retorna o valor de um campo. Cada campo originará uma função, por exemplo, o campo nome originará a função ***GetNome***, que retornará o valor do campo ***Nome***.
- **set*Campo***: função que atribui um valor fornecido como argumento a um campo. Cada campo originará uma função, por exemplo, o campo nome originará a função ***SetNome()***, que atribuirá o valor ao campo ***Nome***.
- **calculaScore**: calcula um score atribuído ao jogador, definido por:

$$4 * Jogos + 6 * Pontos - 0.5 * Faltas - Cartoes.$$

Implemente outro TAD, **`TTime`**. Cada time possui os campos: ***Nome*** (nome do time), ***Jogadores*** (um vetor de jogadores, a posição no vetor indica o número da camisa do jogador), ***Vitorias*** (numero de vitorias do time), ***Derrotas*** (numero de derrotas do time), ***Empates*** (numero de empates do time). Este TAD deverá possuir as seguintes operações:

- **create*TTime***: função que aloca dinamicamente uma variável do tipo TTime.
- **get*Campo*** e **set*Campo***: funções que retornam, ou atribuem, o valor de um campo. Semelhante aos implementados na TAD **`TJogador`**.
- **calculaScore**: calcula um score atribuído ao time em função de seus jogadores. É definido pela média aritmética dos scores dos jogadores.
- **calculaPontos**: calcula os pontos conquistados pelo time, definido por: $3 * Vitorias + Empates$.

Entrada

A entrada começa com quatro inteiros na primeira linha: T, representando o número de times no campeonato; J, representando o número de jogadores inscritos em cada time; N, representando o número de jogadores por partida; e S, representando o número de entradas de súmulas do juiz.

A seguir serão definidos os dados de cada um dos T times e seus J jogadores: na primeira linha será definido o ***Nome*** do time, e na segunda linha serão definidos os ***Nome***'s dos J jogadores deste time, separados por um espaço. Os campos numéricos do time e dos jogadores deverão ser inicializados com o valor padrão 0 (zero). Os jogadores são definidos na ordem de seus números de camisa e devem ser inseridos no vetor de jogadores do seu respectivo time. O vetor de jogadores também deve ser alocado dinamicamente pela implementação da TAD correspondente. Haverá uma linha em branco separando a definição dos dados de cada um dos times com seus jogadores.

Após a definição dos times e jogadores, serão definidas as S entradas da súmula do juiz. Cada linha representa uma entrada da súmula, composta pelo código da entrada e seus respectivos dados, de acordo com o seguinte padrão:

Código	Descrição	Dados
1	Novo jogo	Time1 Time2
2	Escalação	Time Jogador1 Jogador 2 ... Jogador N
3	Ponto marcado	Time Jogador
4	Falta marcada	Time Jogador
5	Cartão aplicado	Time Jogador
6	Fim de jogo	–

Os times são identificados pela ordem de definição na entrada, iniciando pelo valor 0 (como nos índices de um vetor). Os jogadores são referenciados pelo número da camisa. Veja que, ao interpretar os dados das súmulas, os dados de times e jogadores deverão ser atualizados de acordo com as ações relacionadas a cada código:

1. Iniciar variáveis para manipular corretamente os dois times.
2. Incrementar o número de jogos de cada jogador escalado.
3. Contabilizar os pontos marcados pelo jogador no campeonato e o número de pontos marcados pelo time na partida atual.
4. Contabilizar o número de faltas marcadas pelo jogador.
5. Contabilizar o número de cartões recebidos pelo jogador.
6. Finalizar o jogo, contabilizando as vitórias, derrotas e/ou empates para os dois times envolvidos na partida. O vencedor será o time que marcar mais pontos na partida, e o empate ocorre quando o número de pontos das duas equipes forem iguais.

Saída

A saída deve respeitar o seguinte formato:

```
Time campeao: <nome do time> <score do time> <pontos do time>

Time 1: <nome do time 1> <score do time 1> <pontos do time 1>
Time 2: <nome do time 2> <score do time 2> <pontos do time 2>
:
Time T: <nome do time T> <score do time T> <pontos do time T>
```

No caso de empate em pontos, o time que obtiver o maior score será o campeão. Ao persistir o empate, o último time cadastrado deverá ser retornado como campeão.

Exemplo de entrada e Saída

Entrada	Saida
2 7 5 38 Vasco Martin_Silva Andre_Rocha Rodrigo Rafael_Vaz Guinazu Fellipe_Bastos Bernardo Flamengo Felipe Leo_Moura Chicao Samir Caceres Erazo Elano 1 0 1 2 0 0 1 2 4 5 2 1 1 2 3 5 6 3 0 2 3 1 5 3 1 3 3 0 2 3 0 1 4 0 2 4 0 0 4 1 2 4 1 5 5 0 2 5 1 3 6 1 0 1 2 0 2 3 4 5 6 2 1 0 1 3 5 6 3 0 2 3 1 5 3 0 5 3 1 5 4 0 2 4 0 3 4 1 6 4 0 5 5 0 2 6 1 0 1 2 1 0 1 2 4 5 2 1 0 1 3 5 6 3 1 5 3 1 5 3 1 0 4 1 5 4 1 0 5 1 0 6	Time campeao: Flamengo 16.79 4.00 Time 1: Vasco 9.36 4.00 Time 2: Flamengo 16.79 4.00

OBSERVAÇÕES: A definição dos jogadores é feita em uma única linha no arquivo de entrada. Na representação da entrada acima a linha foi quebrada em duas apenas pela falta de espaço para acomodar o texto. É necessária a

implementação do método *main*, que deve permitir uma interação com os métodos definidos e implementados pelas TADs.