

Linguagens de Programação 2021/2022

Exame Época Normal - Duração: 2 horas

Num determinado jogo, um jogador joga contra k adversários diferentes sendo jogadas 3 partidas contra cada adversário. Os resultados obtidos pelo jogador são representados por uma matriz $k \times 3$ cujos elementos são apenas 0, 1 e 3, tal como no exemplo abaixo:

$$\begin{array}{|c|c|c|} \hline 1 & 1 & 3 \\ \hline 0 & 0 & 1 \\ \hline 1 & 3 & 3 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

Isto significa que o jogador em causa defrontou 4 adversários. Contra o primeiro adversário empatou as duas primeiras partidas e ganhou a terceira (primeira linha). Contra o segundo adversário perdeu a primeira e a segunda partida e empatou o terceira (segunda linha) etc.

Crie uma classe `Jogador` para representar e manipular a matriz de resultados de um jogador. A classe deve ter pelo menos um atributo para representar a matriz de resultados e outro que guarde o nome do jogador. Implemente depois:

1. *(3,5 val)* Um construtor que receba três argumentos sendo eles: i) o nome do jogador, ii) o número de adversários com que ele jogou e iii) um vetor com os resultados obtidos pelo jogador contra os adversários com que jogou. Este construtor deverá preencher a matriz de resultados com os valores presentes no vetor recebido. O construtor deve ter em conta todos os tipos de erros que possam ocorrer lançando exceções sempre que necessário.
2. *(1,0 val)* Um método `VerNome` que devolva o nome do jogador;
3. *(2,0 val)* O operador `[]` para aceder aos elementos da matriz de resultados.
4. *(2,5 val)* O operador `<<` para imprimir a matriz de resultados (de acordo com o output abaixo);
5. *(2,0 val)* Um método `Total` para calcular o número total de pontos obtidos pelo jogador (soma dos elementos da matriz de resultados).
6. *(2,0 val)* Um método booleano `Possivel` que receba o número de jogadores que ainda falta defrontar e verifique se, tendo em conta esse número e os pontos já obtidos nos jogos disputados, é possível atingir os 30 pontos.
7. *(2,5 val)* O operador `==` para verificar se dois jogadores têm exatamente o mesmo número de vitórias;
8. *(3,5 val)* A classe `Jogador3` que herda publicamente da classe `Jogador`, sendo específica para matrizes de resultados onde o jogador apenas jogou contra 3 adversários. Esta classe deve ter: i) um construtor pertinente (ver função `main`); e ii) um método `Simetrica` que verifique se a matriz de resultados é simétrica e devolva uma string com essa informação. Nota: uma matriz quadrada é simétrica se cada elemento (i, j) for igual ao elemento (j, i) .
9. *(1,0 val)* Um bloco `try...catch` na função `main` para lidar com todas as exceções anteriores apresentando mensagens de erro específicas.

O programa deve funcionar com a seguinte função main:

```
int main(){
    Jogador R1("Adelino",3,{1, 0, 3, 1, 1, 1, 3, 0, 1});
    Jogador R2("Bernardo",5,{1, 0, 3, 1, 1, 1, 3, 0, 0, 3, 3, 3, 0, 0, 0});
    Jogador R3("Fausto",2,{3, 3, 3, 1, 3, 3});

    cout<<R1.VerNome()<<" ("<<R1.Total()<<" pontos)\n"<<R1 << "\n";
    cout<<R2.VerNome()<<" ("<<R2.Total()<<" pontos)\n"<<R2<< "\n";
    cout<<R3.VerNome()<<" ("<<R3.Total()<<" pontos)\n"<<R3 << "\n";

    if(R2.Possivel(5))
        cout<<"0 "<<R2.VerNome()<<" ainda pode chegar aos 30 pontos.\n";
    else
        cout<<"0 "<<R2.VerNome()<<" ja nao pode chegar aos 30 pontos.\n";

    Jogador3 R4("Osvaldo",{3, 0, 1, 0, 3, 1, 1, 1, 1});
    if(R4==R1)
        cout<<"0 "<<R4.VerNome()<<" e o "<<R1.VerNome()<<" tem as mesmas vitorias.\n";
    cout<<"A matriz R4 "<<R4.Simetrica()<<".\n";

    return 0;
}
```

O output esperado do programa é:

Adelino (11 pontos)

```
| 1 0 3 |
| 1 1 1 |
| 3 0 1 |
```

Bernardo (19 pontos)

```
| 1 0 3 |
| 1 1 1 |
| 3 0 0 |
| 3 3 3 |
| 0 0 0 |
```

Fausto (16 pontos)

```
| 3 3 3 |
| 1 3 3 |
```

0 Bernardo ainda pode chegar aos 30 pontos.
0 Osvaldo e o Adelino tem as mesmas vitorias.
A matriz R4 e simetrica.