

Linguagens de Programação 2021/2022

Exame Época de Recurso - Duração: 2 horas

O *TenShot* é um jogo em que um jogador tenta derrubar 6 latas que se encontram em pirâmide. O jogo é composto por 10 jogadas. Antes de cada jogada, todas as 6 latas são colocadas em pirâmide, independentemente do número de latas que foram deitadas abaixo na jogada anterior. Em cada jogada, o jogador atira uma única bola para derrubar as latas da pirâmide e o número de latas derrubadas corresponderá ao número de pontos do jogador nessa jogada. Isto significa que, no máximo, um jogador poderá obter $10 \times 6 = 60$ pontos no final das 10 jogadas.

Crie uma classe `TenShot` para representar e manipular os resultados obtidos por um jogador no jogo *TenShot*. A classe deve ter um atributo para representar o vetor de pontos do jogador e outro que guarde o nome do jogador. **Em tudo o que implementar deverá lançar todas as exceções que achar pertinente, exceto quando for especificado o contrário.** Implemente:

1. (2,5 val) Um construtor por omissão que **apenas** inicialize o atributo nome com uma `string` vazia e outro que receba o nome do jogador e o seu vetor de pontos.
2. (1,0 val) Um método `VerNome` que devolva o nome do jogador.
3. (2,0 val) Um método `Total` que devolva o número de pontos do jogador.
4. (2,0 val) O operador `[]` para aceder aos elementos do vetor de pontos.
5. (3,0 val) O operador `<<` para imprimir o nome do jogador (deve reservar 9 espaços para o nome do jogador), o seu vetor de pontos e o seu número total de pontos (ver output abaixo):
6. (3,0 val) O operador `>>` para ler informação sobre um jogador na forma

Nome (p₁, p₂, ..., p₁₀)

Assuma que a informação é sempre introduzida no formato correto, pelo que não precisa de fazer quaisquer validações nem lançar exceções neste operador.

7. (2,5 val) Um método `Sequencia` que determine qual a maior sequência de 6 pontos seguidos obtida pelo jogador.
8. (4,0 val) A função global `Ordenacao` que recebe 3 objetos do tipo `TenShot` e os imprime por ordem crescente de pontos num ficheiro com o nome *FinalScore.txt* (ver output abaixo). Esta função deve primeiro verificar se há jogadores com o mesmo número de pontos e, em caso afirmativo, lançar uma exceção. Na implementação desta função é obrigatório o uso do tipo de dados *set*. **A não utilização desta estrutura implica uma redução da cotação da questão em 1,5 valores.**

O programa deve funcionar com a seguinte função main:

```
int main(){
    try{
        TenShot P1("Alberto",{6, 2, 5, 1, 6, 6, 3, 0, 1, 4});
        TenShot P2("Serafim",{6, 6, 1, 2, 1, 6, 6, 6, 0, 5});
        TenShot P3;
        cout<<"Introduza o terceiro jogador ( nome (p1, ... , p10) ):\n";
        cin>>P3; //na execução deve escrever: Aurora (0, 1, 1, 3, 6, 4, 6, 0, 1, 2)

        cout<<"\nInformacao dos jogadores:\n";
        cout<<P1<<"\n"<<P2<<"\n"<<P3<<"\n";

        cout<<"\nMaior sequencia de 6 pontos:\n";
        cout<<"  "<<P1.VerNome()<<": "<<P1.Sequencia()<<"\n";
        cout<<"  "<<P2.VerNome()<<": "<<P2.Sequencia()<<"\n";
        cout<<"  "<<P3.VerNome()<<": "<<P3.Sequencia()<<"\n";

        Ordenacao(P1,P2,P3);
        cout<<"\nOs dados foram escritos no ficheiro.\n\n\n";
        return 0;
    }catch(runtime_error& e){
        cout<<e.what()<<"\n";
    }catch(...){
        cout<<"Erro inesperado\n";
    }
}
```

O output esperado do programa é:

```
Introduza o terceiro jogador ( nome (p1, ... , p10) ):
Aurora (0, 1, 1, 3, 6, 4, 6, 0, 1, 2)
```

Informacao dos jogadores:

```
Alberto: (6, 2, 5, 1, 6, 6, 3, 0, 1, 4) -> 34
Serafim: (6, 6, 1, 2, 1, 6, 6, 6, 0, 5) -> 39
Aurora: (0, 1, 1, 3, 6, 4, 6, 0, 1, 2) -> 24
```

Maior sequencia de 6 pontos:

```
Alberto: 2
Serafim: 3
Aurora: 1
```

Os dados foram escritos no ficheiro.

A informação apresentada no ficheiro *FinalScore.txt* deve ser a seguinte:

```
Ordenacao:
Aurora: 24
Alberto: 34
Serafim: 39
```