



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

2020

Aula 03 – Conceitos Básicos de OO

Atenção

1. As definições das classes usadas nos exercícios encontram-se **disponíveis no e-Disciplinas**.
2. Os nomes, os atributos, os métodos, e as respectivas assinaturas das classes dadas **devem seguir o especificado** em cada exercício para fins de correção automática.
3. A **ordem de declaração** de atributos e métodos fornecidos **não deve ser alterada**. Caso contrário, poderá haver redução automática da nota.
4. A função main **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*.

Exercício 01

Usando a definição fornecida e as especificações a seguir, implemente a classe Equipe:

```
class Equipe {  
public:  
    string nome;  
    int numeroDeMembros = 0;  
  
    int getMaximoDeTorcedores();  
    void imprimir();  
};
```

- Uma Equipe representa um time, que possui um nome e um numeroDeMembros (quantidade de atletas);
- Pensando em uma competição como o InterUSP, onde o espaço para torcedores é limitado, o método getMaximoDeTorcedores() retorna o número máximo de torcedores permitidos para uma equipe. Esse valor é dado pela seguinte equação:

$$\text{número máximo de torcedores} = 15 + 25 * \text{número de membros}$$

- O método imprimir() deve exibir na tela (usando o cout) os dados de cada Equipe no seguinte formato:

<nome> - <número de membros> membros



Sendo <...> o valor armazenado em cada variável. Por exemplo, para um objeto Poli com 22 membros, a saída seria:

Poli - 22 membros

Exercício 02

Usando a definição fornecida e as especificações a seguir, implemente a classe Modalidade:

```
class Modalidade {  
public:  
    string nome;  
    Equipe* e1 = NULL;  
    Equipe* e2 = NULL;  
  
    Equipe* getMaiorTorcida();  
    bool adicionar(Equipe *e);  
  
    void imprimir();  
};
```

- Uma Modalidade é constituída por um nome, e, no máximo, por duas equipes.
- O método getMaiorTorcida() retorna um ponteiro para a equipe com o maior número de torcedores permitidos. Considere que duas equipes nunca terão um número máximo de torcedores igual. Caso só exista uma Equipe, então ela deve ser retornada. Caso não existam equipes na Modalidade, o método getMaiorTorcida() deve retornar NULL.
- O método adicionar(Equipe* e) armazena uma equipe em e1 ou e2 (nesta ordem) e retorna true. Caso a equipe já tenha sido adicionada ou não seja possível adicionar mais equipes (já existem duas equipes na Modalidade) deve-se retornar false.
 - Compare as equipes usando ==. Não adicione, portanto, uma equipe caso o objeto já tenha sido adicionado.
 - Ao testar seu programa **não** atribua equipes diretamente pelos atributos e1 e e2. Use sempre o método adicionar().
- O método imprimir() deve exibir na tela os dados de cada Modalidade no seguinte formato:

<nome da modalidade> - <nome da equipe com maior torcida> com maior torcida

<nome da equipe 1> - <número de membros da equipe 1> membros

<nome da equipe 2> - <número de membros da equipe 2> membros



Por exemplo, no caso da modalidade “Futebol” com duas equipes, “Poli” e “FEA”, teríamos a saída:

Futebol – Poli com maior torcida

Poli – 22 membros

FEA – 17 membros

Na chamada do método `imprimir()`, suponha que sempre existem duas Equipes adicionadas à Modalidade.

Considere que os atributos são sempre atribuídos antes de se chamar os métodos que deles dependem.

Testes do Judge

Exercício 1

- Equipe: Teste `getMaximoDeTorcedores` com nenhum membro
- Equipe: Teste `getMaximoDeTorcedores` com membros
- Equipe: Teste `imprimir`

Exercício 2

- Modalidade: Teste `getMaiorTorcida` com nenhuma equipe na Modalidade
- Modalidade: Teste `getMaiorTorcida` com uma equipe na Modalidade
- Modalidade: Teste `getMaiorTorcida` com duas equipes na Modalidade
- Modalidade: Teste `adicionar` com nenhuma equipe adicionada
- Modalidade: Teste `adicionar` com uma equipe adicionada
- Modalidade: Teste `adicionar` com duas equipes adicionadas
- Modalidade: Teste `adicionar` com a mesma equipe
- Modalidade: Teste `imprimir`