



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

2020

### Aula 4 – Encapsulamento

#### Atenção

- Código inicial a ser usado na resolução dos exercícios encontra-se **disponível no e-Disciplinas**.
- Submeta um arquivo comprimido (faça um “.zip”) colocando apenas os arquivos “.cpp” e “.h”. Não crie pastas no “zip”.
- Ao enviar para o Judge mantenha o `#define NUMERO_MAXIMO_VALORES 10` com o valor 10, por motivos de correção.

Utilize o código fornecido no e-Disciplinas para implementar as classes **Equipe** e **Modalidade**, aplicando os conceitos de encapsulamento vistos em aula.

#### Exercício 1

No arquivo main.cpp é fornecida a definição e a implementação da classe Equipe. Separe-a em dois arquivos, “**Equipe.h**” e “**Equipe.cpp**”. O arquivo .h deve conter apenas a definição. O arquivo .cpp deve conter apenas a implementação. Use adequadamente as diretivas de compilação.

Defina a visibilidade dos atributos e dos métodos de modo que os atributos sejam acessíveis apenas no escopo da classe e os métodos sejam acessíveis externamente.

#### Exercício 2

Implemente a classe Modalidade, definida a seguir. **Inclua os arquivos “.h” e “.cpp” referentes a essa classe ao projeto do Code::Blocks** (clique com o botão direito no projeto e selecione “Add files..”). Note que essa classe utiliza a classe Equipe, do Exercício 1. Novamente, é necessário separar a classe em dois arquivos, “**Modalidade.h**” e “**Modalidade.cpp**”, usando adequadamente as diretivas de compilação. Defina corretamente a **visibilidade** de seus métodos e atributos.

```
#define NUMERO_MAXIMO_VALORES 10

class Modalidade {
    int calculaTotalTorcida();
    bool adicionar(Equipe *e);

    string getNome();
    int getQuantidade();
    void setNome(string nome);

    void imprimir();
};
```



A implementação da classe Modalidade deve atender aos seguintes requisitos:

- O método `calculaTotalTorcida` deve retornar o somatório do número máximo de torcedores de cada equipe na modalidade:  
Caso não existam equipes adicionadas à modalidade, esse método deve retornar -1.
  - **Dica:** Use o método da classe Equipe.
- O método `adicionar` deve adicionar, se possível, o objeto do tipo Equipe passado como parâmetro ao vetor **equipes** (crie um atributo para isso). Caso o vetor já esteja completamente preenchido, o método não modifica o vetor e retorna **false**. Caso seja bem sucedido, deve retornar **true**. Utilize a constante `NUMERO_MAXIMO_VALORES` como o número máximo de equipes que a modalidade comporta. Controle a quantidade de objetos adicionados com um atributo quantidade.
- O método `getQuantidade` retorna o número de equipes que foram adicionadas ao vetor **equipes**, retornando o valor do atributo quantidade.
- O método `imprimir` deve mostrar na tela as informações de cada modalidade, seguindo o formato:

`<nome> - <numero total de de torcedores> torcedores no maximo`

Além de imprimir as informações de todas as equipes que foram adicionadas à modalidade. Por exemplo:

InterUSP - 555 torcedores no maximo  
Poli - 10 membros  
FEA - 11 membros

**Dica:** a classe Equipe já possui um método `imprimir` – utilize-o para facilitar a implementação deste método!

**Observação:** esse método não será corrigido pelo Judge. Utilize-o para testar a sua classe!

Crie um `main` que utilize as classes Equipe e Modalidade, a fim de testar as implementações.

Lembre-se de utilizar a diretiva `#ifndef` nos arquivos de cabeçalhos (".h") para evitar problemas de conflitos de definição de classes causados por múltiplas inclusões de um cabeçalho.

Dicas importantes

1. Os nomes dos métodos e das classes **devem seguir exatamente o especificado** para fins de correção automática.
2. A função `main` não deve ser submetida. Caso contrário, a correção automática retornará um *Compilation Error*.

## Testes do Judge

### Exercício 1

- Teste dos *setters* e dos *getters*
- Teste `imprimir`



Exercício 2

- Teste dos *setters* e dos *getters* da Modalidade
- Teste adicionar com vetor vazio
- Teste adicionar com vetor parcialmente preenchido
- Teste adicionar com vetor cheio
- Teste calculaTotalTorcida com vetor vazio
- Teste calculaTotalTorcida com vetor parcialmente preenchido
- Teste calculaTotalTorcida com vetor cheio