



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

2020

Aula 09 – Programação defensiva

Atenção

1. Código inicial para resolução dos exercícios encontra-se disponível no e-Disciplinas.
2. Para a resolução dos exercícios, adicione atributos privados às classes conforme necessário, desde que se mantenha as assinaturas e funcionamento especificados no enunciado.

Exercício 01

Altere o código fornecido da classe **Modalidade** seguindo as seguintes características:

1. O construtor deve jogar uma exceção do tipo `invalid_argument` (da biblioteca padrão) caso o valor de `maximoEquipes` seja menor ou igual a 1 ou caso a string `nome` seja vazia.
2. O método `adicionar` deve:
 - Não ter valor de retorno (ser `void`).
 - Gerar uma exceção do tipo `overflow_error` (da biblioteca padrão) caso o vetor `equipes` esteja cheio.

Dicas: Uma string vazia é dada por `""`. Não se esqueça de incluir o cabeçalho `stdexcept`.

Exercício 02

Crie e implemente a exceção `EquipeRepetida` de modo que:

- Seja uma classe filha de `invalid_argument` (da biblioteca padrão).
- A classe contenha apenas seu destrutor e construtor, o qual recebe como argumento uma mensagem (string), como exposto em aula.

Modifique a classe **Modalidade** fornecida para que o método `adicionar` gere um erro do tipo `EquipeRepetida` caso tente-se adicionar uma **Equipe** previamente adicionada – isto é, o mesmo objeto (compare-o com `==`).

Exercício 03

Altere a classe **Competição**, utilizando comandos `try-catch` dentro desta classe, de forma que o método `adicionar(Modalidade *m)`, após adicionar a modalidade no vetor (já implementado),



adiciona todas as **Equipes** que estão no vetor equipes de **Competição** na modalidade m. Para isso, deve-se usar o método adicionar de **Modalidade** implementado anteriormente. Caso ao chamar o método adicionar de **Modalidade** aconteça uma exceção EquipeRepetida, capture e destrua a exceção, mas continue adicionando o restante das **Equipes** da **Competição** na **Modalidade**. (Não se preocupe com o overflow_error).

Por exemplo, suponha que a **Competição** InterUSP tenha duas **Equipes**: Poli e FEA. Considere uma **Modalidade** futebol, que já tem a **Equipe** Poli adicionada. Ao adicionar a **Modalidade** futebol à **Competição** InterUSP, a Modalidade futebol ficará com as **Equipes** Poli e FEA, além de ser adicionada ao vetor de modalidades da **Competição**.

Testes do Judge

Exercício 1

- Modalidade: construtor e adicionar com valores válidos
- Modalidade: nome inválido
- Modalidade: maximoEquipes inválido
- Modalidade: adicionar com vetor cheio

Exercício 2

- EquipeRepetida é filha de invalid_argument
- Modalidade: adicionar Equipe repetida
- Modalidade: adicionar Equipe não repetida

Exercício 3

- Competicao: adicionar uma Modalidade sem nenhuma Equipe repetida
- Competicao: adicionar uma Modalidade com Equipes repetidas