



PUC Minas

Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de Software - Campus Coração Eucarístico
Programação Modular
Prof. Daniel Kansaon

Valor: 5 pontos

OBS: Esta atividade pode ser feito em duplas

Entrega: especificado no canvas (23:59)
com o esquema UML final

Formato de Envio: Projeto Java zipado + PDF

Entrega em atraso: Entregas em atraso **não serão** consideradas.

Sistema de Gerenciamento de Hackathons Universitários

Você foi convidado para participar do projeto de um sistema para gerenciar um **Hackathon Universitário**, promovido anualmente por várias universidades. Dentre as várias funcionalidades que o sistema deve oferecer, você foi contratado principalmente para construir a parte responsável por gerenciar os Projetos e as Equipes participantes.

Considere o diagrama UML abaixo como parte inicial da solução. É preciso pensar em atributos e métodos para cada classe modelada. Todo o sistema será gerenciado, a princípio, em memória principal. Dessa forma, implemente classes auxiliares para controlar as coleções de objetos do sistema. Por exemplo, crie uma classe **Equipes** para manter a coleção de objetos do tipo **Equipe** e uma classe **Apresentacoes** para representar objetos do tipo **Apresentacao**. Essas classes de coleção devem oferecer métodos para incluir e pesquisar elementos, e devem seguir o padrão de projeto **Singleton**, garantindo que apenas uma instância da coleção seja utilizada ao longo do sistema.

Exemplo no main (**não é preciso criar menu**):

- Cria 2 equipes com 5 alunos cada;
- Adiciona um projeto diferente para cada equipe;
- Criar uma banca avaliadora para cada projeto (cada banca deve conter 4 jurados);
- Calcula a nota final de cada projeto a partir da apresentação (considere a média das notas dos jurados);
- Lista os projetos com nota final ≥ 7 (usar stream).

Critérios para a avaliação:

1. Implementação correta do main() e do diagrama UML completo - **2 pontos**
2. Uso correto do padrão Singleton nas classes de coleção (Equipes e Apresentacoes) e sua aplicação no main() - **1 ponto**
3. Utilização da estrutura proposta (**Figura 1**) com os métodos e atributos necessários - **1 ponto**
4. Organização geral do código: encapsulamento, nomes claros, exemplo funcional no main() com listagem de projetos aprovados - **1 ponto**

Figura 1. Esboço das Classes do Problema

