

2022_1 - PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II - TA_TN - METATURMA

PAINEL > MINHAS TURMAS > 2022_1 - PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II - TA_TN - METATURMA > GERAL
> L02E07 - DIÁRIO CLASSE (2,0 PTS)


 Descrição

 Enviar

 Editar

 Visualizar envios

L02E07 - Diário classe (2,0 pts)

 **Data de entrega:** sexta, 1 Jul 2022, 23:59

 **Arquivos requeridos:** DiarioClasse.cpp, TesteDiarioClasse.cpp ( [Baixar](#))

Tipo de trabalho:  Trabalho individual

O objetivo desse exercício é praticar a escrita de **Testes Unitários** utilizando o framework **doctest**. Além disso, também será possível ter contato com a metodologia **TDD (Test Driven Development)**.

Já foram implementados 4 testes básicos no arquivo main.cpp (nem todos passam inicialmente, essa é a ideia do TDD), e você deve implementar mais **10** outros testes (no arquivo TesteDiarioClasse.cpp). Coloque apenas **uma asserção** por caso de teste (semelhante aos testes já feitos).

Um código base para a classe DiarioClasse é fornecido e você deve terminar de implementá-la. Essa classe possui um único método, com a assinatura abaixo:

- `bool determinarAprovacao(double notaSemestre, double notaEspecial, double frequencia);`

Esse método é responsável por informar se um estudante foi aprovado ou não na disciplina. Os parâmetros `notaSemestre` e `notaEspecial` representam a nota final no semestre e a nota obtida no exame especial, respectivamente, e ambas devem estar no domínio **[0, 100]**. O parâmetro `frequencia` varia entre 0 e 1, e representa o percentual de frequência no semestre.

Os critérios para aprovação/reprovação são detalhados abaixo:

- Aprovação:**

Será considerado aprovado o aluno que obtiver, simultaneamente, no mínimo, 60 pontos no semestre e, no mínimo, 75% de frequência nas atividades acadêmicas em que se matriculou no semestre letivo.

- Reprovação:**

Será considerado reprovado o aluno que obtiver nota inferior a 60 pontos ou for infrequente.

O aluno somente poderá fazer o **Exame Especial** se tiver obtido o **conceito E** ($40 \leq \text{notaSemestre} < 60$) e for **frequente** ($\text{frequencia} \geq 0.75$). Nesse caso, se tirar no mínimo 60 pontos será aprovado.

Por fim, o seu método também deverá tratar situações excepcionais, lançando as seguintes exceções (já implementadas no arquivo `DiarioClasse.hpp`):

- ExcecaoNotaSemestreInvalida:** Caso o parâmetro `notaSemestre` informado esteja fora do intervalo válido.
- ExcecaoNotaEspecialInvalida:** Caso o parâmetro `notaEspecial` informado esteja fora do intervalo válido. Também deve ser lançada se `notaEspecial` foi informada mas o aluno não estava apto para fazer o Exame Especial.
- ExcecaoFrequenciaInvalida:** Caso o parâmetro `frequencia` informado esteja fora do intervalo válido.

Atenção, não utilize a instrução `#define DOCTEST_CONFIG_IMPLEMENT_WITH_MAIN` em nenhum dos arquivos! Também será verificada uma **cobertura de pelo menos 70%** (declarações e decisões). Você pode checar a cobertura do seu código com o comando: `gcovr -r . --filter="DiarioClasse.cpp" -s`. Lembrando que o código deve ser compilado com a flag `--coverage`. Essa informação também é exibida na aba 'Compilação' do ambiente de desenvolvimento do Moodle.

Dica 1:

O código com todos os arquivos necessários para o exercício pode ser copiado [aqui](#), caso você queira fazer localmente.

Referências:

<https://github.com/doctest/doctest>

Arquivos requeridos

DiarioClasse.cpp

```
1  #include "DiarioClasse.hpp"
2
3  bool DiarioClasse::determinarAprovacao(double notaSemestre, double notaEspecial, double frequencia) {
4
5      if (notaSemestre < 0)
6          throw ExcecaoNotaSemestreInvalida();
7
8      if (notaSemestre > 60)
9          return true;
10     else
11         return false;
12
13 }
```

TesteDiarioClasse.cpp

```
1  #include "doctest.h"
2  #include "DiarioClasse.hpp"
```

[VPL](#)[◀ L02E06 - Fila Excepcional \(3,0 pts\)](#)

Seguir para...

[PF - Enunciado ▶](#)