

Trabalho Avaliativo - Sistema de Avaliação de Notas

Trabalho em dupla

Entrega: 28/02/2025

Enunciado

Desenvolva um sistema de avaliação de notas que calcula e classifica o desempenho de alunos de acordo com sua nota final. O sistema deve avaliar e retornar uma mensagem condizente com o desempenho do aluno, levando em consideração as seguintes faixas de notas e suas respectivas classificações:

1. **Nota ≥ 90 :** Excelente
2. **Nota ≥ 75 e < 90 :** Bom
3. **Nota ≥ 50 e < 75 :** Regular
4. **Nota ≥ 30 e < 50 :** Insuficiente
5. **Nota < 30 :** Reprovado

O sistema deve ser capaz de:

- Receber a nota final de um aluno.
- Retornar uma classificação com base na faixa de nota do aluno.
- Exibir uma mensagem apropriada de acordo com a situação do aluno.

Além do código da aplicação, você deve criar um código de testes automatizados para validar o comportamento do sistema em cada uma das faixas de nota. O sistema também deve conter um plano de testes com os casos de teste utilizados, incluindo:

- Descrição de cada teste.
- Entradas para cada teste.
- Resultado esperado.

Requisitos Técnicos

1. **Linguagem de Programação:** JAVA.
2. **Estrutura do Sistema:**
 - Crie uma classe estudante que contenha os atributos do aluno, como nome e notaFinal.
 - Crie uma classe avaliadorDeNotas que seja responsável por calcular a classificação do aluno com base na sua nota final.
 - O sistema deve ser modular, com funções e métodos bem definidos.

3. Testes Automatizados:

- Use a ferramenta de testes automatizados JUnit.
- Os testes devem verificar se as diferentes faixas de nota estão sendo tratadas corretamente.

4. Plano de Testes: Forneça um plano de testes detalhado (em inglês), que cubra todos os cenários possíveis de entrada e as saídas esperadas do sistema.

Entrega

A entrega será pelo seguinte link: [Trabalho UC11 · DalvanaRibeiro/UC11_241N · Discussion #1](#) deve conter:

1. Nomes e o link do repositório
2. O código-fonte completo da aplicação.
3. O código de testes automatizados.
4. O plano de testes em inglês, detalhando todos os cenários e expectativas.
5. Qualquer documentação que ajude a entender o funcionamento do sistema.

CrITÉrios de Avaliação

- **Funcionalidade:** O sistema deve calcular e classificar corretamente as notas, retornando a mensagem correta para cada faixa.
- **Qualidade do Código:** O código deve ser bem estruturado, com boa utilização de funções/métodos e lógica clara.
- **Testes Automatizados:** Todos os cenários de teste devem ser cobertos. Os testes devem ser escritos de forma que testem efetivamente todas as faixas de nota e suas respectivas classificações.
- **Plano de Testes:** O plano de testes deve ser completo e bem estruturado, em inglês, com descrição clara de cada caso de teste.
- **Documentação:** O código deve ser bem documentado, com comentários explicativos que ajudem a entender a lógica de implementação.

Exemplo de Fluxo de Sistema

- Entrada: **Nota do aluno = 85**
 - Saída: **"Bom"**
 - Mensagem: **"Seu desempenho foi bom! Continue assim!"**
- Entrada: **Nota do aluno = 95**
 - Saída: **"Excelente"**

- Mensagem: "**Parabéns! Você está com desempenho excelente!**"
- Entrada: **Nota do aluno = 45**
 - Saída: "**Insuficiente**"
 - Mensagem: "**Você não passou, precisa melhorar. Não desista!**"
- Entrada: **Nota do aluno = 20**
 - Saída: "**Reprovado**"
 - Mensagem: "**Você foi reprovado. Busque apoio e estude mais para a próxima!**"

Dicas para Desenvolvimento

1. **Validação de Entrada:** Lembre-se de validar as entradas. O sistema não deve aceitar notas negativas ou superiores a 100.
2. **Testes de Limite:** Certifique-se de criar testes para os limites das faixas de notas, como 90, 75, 50 e 30, para garantir que o sistema está classificando corretamente.
3. **Mensagens de Erro:** Implemente mensagens de erro para entradas inválidas (por exemplo, nota abaixo de 0 ou acima de 100).
4. **Boas Práticas de Programação:** Utilize boas práticas de programação, como a separação de responsabilidades e o uso de funções e métodos claros.