Conteúdo do exercício

Parte superior do formulário

**Instruções para a Atividade Prática 2**

**Visão Geral:**

Esta atividade é uma oportunidade para você aplicar os conceitos aprendidos em sala de aula de forma criativa e prática. O tema é livre, permitindo que você explore suas áreas de interesse dentro do universo da programação Java.

**Formato e Entrega:**

* + **Pontuação:** A atividade vale até 2,0 pontos.
  + **Modalidade:** Pode ser realizada individualmente ou em dupla.
  + **Prazo de entrega:** 26/04/24, até as 23h30.
  + **Ferramentas:** Crie o projeto Java chamado "AtividadePratica2" usando o VS Code, IntelliJ, ou a IDE de sua preferência.
  + **Repositório:** O projeto deve ser subido em um repositório público no GitHub ou plataforma similar.
  + **Colaboração:** Se realizada em dupla, é essencial que ambos os membros contribuam ativamente, evidenciado por commits de cada um no repositório. A nota será atribuída individualmente com base na contribuição.
  + **Link de Submissão:** Após concluir, postar o link de acesso ao repositório aqui. Certifique-se de que o link está correto; erros de link não serão passíveis de revisão.
  + **Tentativas:** Permitida apenas uma tentativa de envio. Certifique-se de revisar seu trabalho cuidadosamente antes de enviar.

**Integridade Acadêmica:**

* + **Recursos:** O uso de materiais da sala, consultas à internet e ao ChatGPT é permitido. No entanto, cópias integrais de código do ChatGPT ou de aulas anteriores não serão aceitas.
  + **Originalidade:** Trabalhos que pareçam substancialmente diferentes do conteúdo discutido em sala podem ser submetidos a uma defesa de código para validação da nota.

**Requisitos Funcionais:**

* + **Estrutura do Código:**
  + A classe principal deve conter apenas o método **main**.
  + O método **main** deve chamar apenas um método responsável pela execução do projeto.
  + Classes do tipo "model" não devem ter saída de dados; toda a saída de dados deve ocorrer em uma classe separada chamada **Sistema.java**, através de métodos estáticos chamados pelo método **main**.
  + **Práticas de POO:**
  + Implementação obrigatória de orientação a objetos, herança e uso de **ArrayLists**.
  + Criar ao menos três subclasses a partir de uma superclasse. Por exemplo, **Funcionario** como superclasse e **Professor**, **Diretor**, **Coordenador** como subclasses.
  + Manter uma classe com um **ArrayList** estático como a única referência à lista de objetos do projeto, com o tipo de **ArrayList** sendo da classe mais abstrata.
  + **Manipulação de Dados:**
  + Métodos para gerenciar dados na lista, incluindo: cadastrar, buscar, excluir, e listar todos, além de excluir todos os objetos da lista. Estas opções devem contemplar os diferentes tipos de objetos (ex: o sistema deve permitir cadastrar, buscar e excluir, Professores, Coordenadores e Diretores, e listar todos os funcionários e excluir todos de uma vez também)
  + **Interface de Usuário:**
  + Apresentar um menu interativo com opções para Cadastro, Busca, Exclusão, Listagem de todos os itens, Exclusão total e Sair.
  + O programa deve executar ações baseadas na opção escolhida e exibir o menu novamente após cada operação, exceto pela opção de saída (0), que finaliza o programa.
  + Mensagens de erro devem ser mostradas para entradas inválidas que não correspondam às opções do menu.

**Avaliação e Feedback:**

Cada projeto será avaliado com base na correção técnica, design do código, e inovação na aplicação dos conceitos.

Parte inferior do formulário