

Trabalho Final – Desabafe
Programação Orientada a Objetos – 2018.2
Prof. Anderson Lemos

Nome: João Teixeira do Nascimento Mat.: 422225

1.Objetivo

Desenvolver um bate-papo entre duas pessoas para desabafar sobre os problemas que está enfrentando e orientação de um profissional para acompanhar melhor o caso da primeira pessoa.

Ao final da conversa os dois participantes dão uma nota para o outro, assim o sistema sempre se mantém com pessoas de boa índole.

2.Classes

2.1.1 – Usuário e suas classes filhas

A classe usuario serve para ser uma classe mãe abstrata de Novato e Master, eles herdam os atributos da classe mãe e preenchem com o seus dados que só se diferenciam no atributo tipo. Dependendo do tipo o usuário ou é Novato ou é Master, isso é decidido após uma conversa terminar a classe 'RepositorioDadosImpl' olha se o usuário que estar recebendo a nota tem mais de 5 avaliações e se ele tiver uma média acima de 4 vira Master senão continua Novato.

2.1.2 – Conversa, mensagem e invite

A classe conversa possui uma lista de mensagem que só possui um atributo mensagem, além disso conversa possui os atributos correspondentes a id dos usuários da conversa. Invite possui um atributo para com a id de quem solicitou a conversa e quem vai receber a solicitação e aceitar ou não.

2.2 – Atores

Os atores do sistema são Novato e Master que são classes filhas de usuario.

Como usuario existe para dar seus atributos de herança para Novato e Master, ela é uma classe abstrata com os atributos: id(int), login(String), senha(String), nome (String), idade(int), ponto(double), numPonto(int), tipo(int), historia(String) e tema(String).

O único atributo que diferencia Novato de Master é o tipo = 1, tipo = 2 respectivamente, o repositório 'RepositorioDadosImpl' fica responsável por manipular listas de usuario e arquivos que armazenem esses atributos.

Novato possui as opções de:

- Cadastrar-se
- Login
- BuscarConversas
- Enviar mensagem
- Sair da conversa
- Encerrar conversa

Master possui as opções de:

- Login
- Aceitar ou não convite para conversar
- Enviar mensagens
- Sair da conversar
- Encerrar Conversa

2.3 Repositório

Os repositórios tanto pegam os dados dos arquivos ao início do código com armazenam nos mesmos arquivos, seja para atualizar uma informação, adicionar ou remover.

O RepositorioDadosImpl fica responsável pelos dados pessoais dos usuários(até mesmo o cálculo de nota do usuário) enquanto que o RepositorioInviteImpl fica responsável por armazenar o convite feito pelo Novato para um Master, a classe mantém as informações só até obter uma resposta de Master, seja qual for. A classe RepositorioListaConversaImpl fica responsável por armazenar a id de quem está conversando e mantém os dados até que o último participante tenha dado a nota para quem conversou com ele. RepositorioConversaImpl fica responsável por armazenar a conversa entre dois usuários e mostrar todas as mensagens quando um deles voltar para o bate-papo.

3. A interface

Existe somente duas interfaces para os dois tipos de usuário. São elas:

Quando inicia o programa:

Bem vindo ao DESABAFE, onde você desabaja sobre os seus problemas com outra pessoa!

Digite 'cadastrar' para realizar o seu cadastro ou 'logar' para entrar na sua conta!

E quando começa uma conversa:

Digite 'mensagem' para escrever uma mensagem ou 'sair' para sair do programa ou 'terminar' para terminar a conversa.

3. Considerações Finais

As equipes serão formadas por no máximo 2 (DOIS) alunos.

Todos os requisitos do trabalho são OBRIGATÓRIOS, e devem ser preenchidos com coisas que façam sentido. Cada requisito não cumprido acarretará na diminuição da nota. As notas serão diferenciadas por aluno no momento da apresentação do trabalho em sala.

A nota do trabalho fará parte da nota final da disciplina, valendo um terço da nota total. A O trabalho deverá ser implementado em JAVA.

4. Dicas

1 Sejam criativos! Façam um sistema que vocês gostariam de usar. Funcionalidades a mais serão muito bem vistas, por exemplo:

Salvar os dados em arquivos para não serem perdidos quando o programa for fechado e serem carregados quando o programa for aberto.

Interface gráfica.

Se seu sistema utiliza alguma mídia (música, video), executar a mídia.

Alguma coisa utilizando computação paralela (Threads).

A interpretação do trabalho é uma forma de avaliação. Se você acha que algo não ficou bem explicado, tente entender como seria a maneira certa de fazê-lo e resolva o problema. Esse é um problema que vocês vão ter que lidar diariamente no mercado de trabalho.

Deixem a parte mais difícil por último, concentrem-se em implementar as classes de acordo com a orientação a objetos. As funcionalidades podem ser implementadas depois.

Procurem dividir o trabalho entre os integrantes.

Sintam-se livres para implementar da forma que for mais conveniente. No entanto, respeitem a OO e convenções de código. Tornem o programa funcional.

Façam o trabalho por partes: leiam com cuidado este documento e vão implementando à medida que forem entendendo.

Tirem dúvidas com o professor, entre vocês e pela Internet.

Utilizem os conceitos vistos em sala: herança, polimorfismo, abstração, encapsulamento, tratamento de erros, coleções, classes abstratas, etc