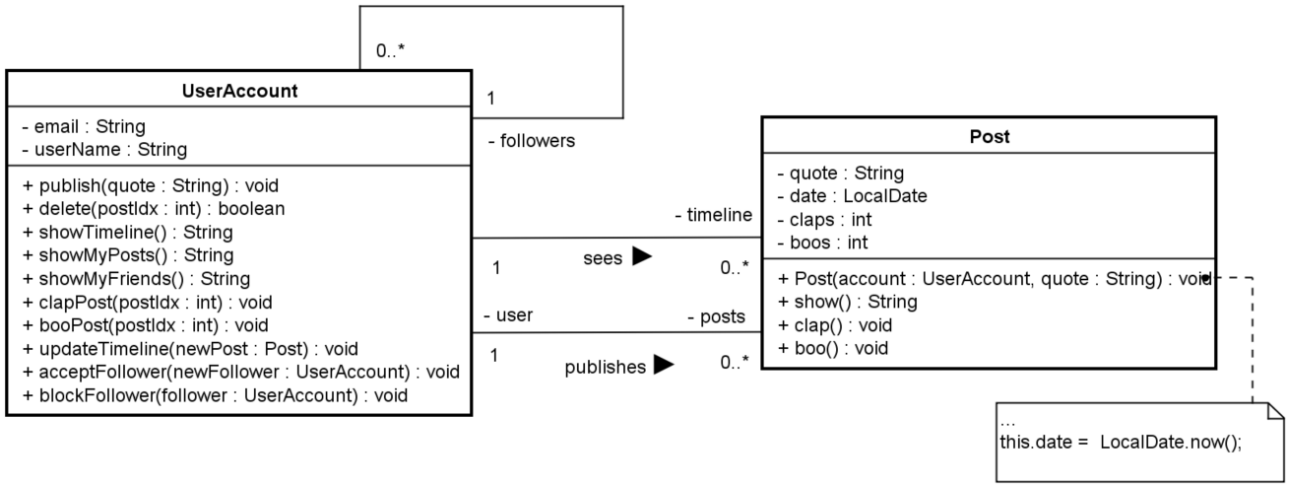
Prontuário: | S | C | | | | | | | - | | Data: 18/09/2019 Término: ___h___ Nota: _____

✓ O exercício é individual e sem consulta. ✓ Não é permitido utilizar qualquer código implementado previamente. ✓ Atribui-se nota zero ao exercício em desacordo com os itens acima.	✓ Crie o seu projeto com o seguinte nome: PRONTUARIO_P1, com SC, sem traço. ✓ Compacte o projeto e chame o professor para recolher o arquivo. ✓ Enviar via Moodle.

Durante o terceiro semestre do curso de ADS você percebeu uma grande oportunidade de negócio: um aplicativo para compartilhamento de mensagens motivacionais (como as que usuários de redes sociais concorrentes já postam, mas sem a foto narcisista junto). Usuários no aplicativo podem realizar (e remover) postagens, aceitar (e bloquear) seguidores e conferir em sua timeline as mensagens inspiradoras postadas por usuários que o aceitaram como seguidor. Um usuário pode também aplaudir uma postagem ou vaiá-la. Além disso, é possível listar suas próprias postagens, as postagens de sua timeline e seus seguidores. Para investigar a viabilidade do projeto, inicialmente, você deverá implementar as classes do domínio em formato console. Considere a especificação a seguir:



Execute as atividades a seguir para a implementação do exercício em Java. Marque com um "✓" as atividades que considera ter concluído, com o "–" as que considera não ter concluído completamente ou com um "**X**" as não realizadas. ATENÇÃO: não indicar com "✓" ou "–" uma atividade não iniciada (será retirada da nota final a pontuação da atividade indicada incorretamente).

#	Descrição	Pontos	Executado
1	Crie o projeto, a classe Principal e as classes do modelo.	1pt	
2	Implemente os atributos das classes do modelo aplicando o modificador de acesso mais adequado segundo o conceito de encapsulamento. Crie métodos de acesso para as classes do modelo segundo o esperado pelo conceito de encapsulamento.	1pt	

3	Estruture os relacionamentos das classes do modelo. Para relacionamentos do tipo 1 para 0..*, utilize arranjos. O arranjo para representar a timeline deverá comportar até 10 Posts. Os demais poderão ter um tamanho suficientemente grande para que a capacidade máxima seja desconsiderada.	1pt	
4	Implemente os métodos da classe Post seguindo a especificação. O método show produz uma String no formato *[<date>] <user.Name> says "<quoteText>" | Claps: <number> | Boos: <number>*. Os métodos clap e boo adicionam, respectivamente, um aplauso ou uma vaia. Não é possível ajustar diretamente o número de aplausos ou vaias, apenas usando os métodos clap e boo.	1,5pt	
5	Desenvolva o método publish da classe UserAccount, que além de incluir um Post no conjunto de posts do usuário, também o adiciona no conjunto de posts da timeline de todos os seguidores desse usuário. Obs: esse método deve ser implementado junto com o método do Item 6, descrito a seguir.	1pt	
6	Implemente o método updateTimeline da classe UserAccount, que adiciona na timeline do usuário um post publicado por alguém que ele segue. Se a timeline estiver cheia, a postagem é colocada no lugar da mais antiga. Uma timeline não necessariamente precisa estar ordenada pela data da postagem.	1pt	
7	Implemente o método delete da classe UserAccount, que remove um post da lista de posts de um usuário. O post removido não precisa ser retirado imediatamente da timeline dos seguidores, ele será removido à medida que postagens mais atuais cheguem a timeline. Use o índice do array para identificar a postagem que será removida.	1pt	
8	Crie os métodos clapPost e booPost da classe UserAccount, de forma a permitir que um usuário aplauda ou vaie uma postagem em sua timeline. Use o índice do array para identificar a postagem que receberá a reação do usuário. Um usuário pode aplaudir ou vaiar mais de uma vez uma mesma mensagem motivacional.	0,5pt	
9	Desenvolva os métodos acceptFollower e blockFollower, que aceita um seguidor em um usuário ou o bloqueia, respectivamente. Usuários bloqueados não receberão novas frases motivacionais do bloqueador, embora mantenham as já existentes na timeline enquanto novas não chegam.	1pt	
10	Adicione um método main à classe Principal e teste as funcionalidades do sistema, tais como a criação e associação entre usuários, novas postagens, claps e boos, etc.	1pt	

O aluno que terminar todas as atividades corretamente primeiro ganha 1pt adicional para usar em outros exercícios.

***** Boa sorte! *****