

DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Estruturas lineares no contexto de orientação a objetos](#) / [\[EP\] Lista de mensagens](#)

Descrição	Visualizar envios
-----------	-----------------------------------

[EP] Lista de mensagens

Data de entrega: segunda, 26 Jul 2021, 23:59
Arquivos requeridos: ListaMensagens.java, Mensagem.java ([Baixar](#))
Tipo de trabalho: Trabalho individual

O professor ABC está desenvolvendo um programa de chat, mas falta implementar algumas classes. Neste exercício, implemente as classes ListaMensagens, Mensagem, MensagemTexto e MensagemImagem (todas as classes são do pacote **chat**).

[acesso public] classe abstrata **Mensagem**:

- **like()** - incrementa em uma unidade a quantidade de likes (quando a classe é instanciada, a quantidade de likes é inicializada com zero).
- **responder(resposta: Mensagem)** - recebe uma instância de Mensagem e faz uma ligação entre a mensagem atual e a resposta de forma que seja possível identificar que a mensagem é resposta da mensagem atual.
 - Por exemplo, assuma que há uma instância de mensagem A com o texto "Quanto eh 1 + 1?" e outra instância de mensagem B com o texto "O resultado eh 2.". Nesse caso, a mensagem B será a resposta de A. Então, o programa poderia realizar a seguinte chamada: msgA.responder(msgB); // Nesse caso msgA e msgB seriam as instâncias com as mensagens A e B respectivamente. Dessa forma, msgA foi respondida com msgB.
- **getRepresentacao(): String** - método abstrato que retorna uma representação em String da mensagem (a representação depende da subclasse)

[acesso package] classe **MensagemTexto** (subclasse de Mensagem):

- **MensagemTexto(texto: String)** - construtor que armazena o texto da mensagem na instância atual.
- **getRepresentacao(): String** - retorna uma String no seguinte formato: "<texto> (likes=<qtd>)"
 - Se a mensagem atual for uma resposta a outra mensagem, a String deve ter o seguinte formato: "<texto> RESPOSTA A [<getRepresentacao() da mensagem respondida sem a parte dos likes>] (likes=<qtd>)"
 - Observações: <texto> é o texto da mensagem, <qtd> é a quantidade de likes da resposta.
 - Exemplo 1: para o texto "teste2" que é uma resposta à mensagem "teste1", a representação será: "teste2 RESPOSTA A [teste1] (likes=0)" (a quantidade de likes é a referente à resposta, nesse exemplo, "teste2" é o texto da resposta).
 - Exemplo 2: para o texto "teste2" que é uma resposta à imagem "figura1 10x10", a representação será: "teste2 RESPOSTA A [figura1 10x10] (likes=0)" (a quantidade de likes é a referente à resposta, nesse exemplo, "teste2" é o texto da resposta).

[acesso package] classe **MensagemImagem** (subclasse de Mensagem):

- **MensagemImagem(arquivomagem: String, largura: int, altura: int)** - construtor que armazena o arquivo da imagem e suas dimensões (largura e altura) na instância atual.
- **getRepresentacao(): String** - retorna uma String no seguinte formato: "<arqlmg> <largura>x<altura> (likes=<qtd>)"
 - Se a mensagem atual for uma resposta a outra mensagem, a String deve ter o seguinte formato: "<arqlmg> <largura>x<altura> RESPOSTA A [<getRepresentacao() da mensagem respondida sem a parte dos likes>] (likes=<qtd>)"
 - Observações: <arqlmg> é o arquivo da imagem, <largura> é a largura da imagem, <altura> é a altura da imagem, <qtd> é a quantidade de likes da resposta.
 - Exemplo 1: para a imagem "img2 20x20" que é uma resposta à mensagem "teste1", a representação será: "img2 20x20 RESPOSTA A [teste1] (likes=0)" (a quantidade de likes é a referente à resposta).
 - Exemplo 2: para a imagem "img2 20x20" que é uma resposta à imagem "figura1 10x10", a representação será: "img2 20x20 RESPOSTA A [figura1 10x10] (likes=0)" (a quantidade de likes é a referente à resposta).

[acesso public] classe **ListaMensagens**: essa classe gerencia uma lista de mensagens

- **ListaMensagens()** - construtor sem parâmetros da classe ListaMensagens.
- **adicionarMensagem(mensagem: Mensagem)** - adiciona mensagem ao final da lista de mensagens.

- **responderMensagem(indiceMensagemOriginal: int, resposta: Mensagem)** - responde a mensagem no índice especificado e adiciona a resposta ao final da lista de mensagens.
- **getMensagens(): Mensagem[]** - retorna vetor com as mensagens da lista. O comprimento do vetor retornado deve ser igual à quantidade de mensagens adicionadas na lista. Observe que a lista de mensagens é composta pelas mensagens originais, assim como pelas mensagens resposta.
- **getMensagensString(): String[]** - funciona da mesma forma que o método getMensagens, mas ao invés de retornar instâncias de Mensagem, retorna uma lista de Strings. Cada String é o retorno obtido por getRepresentacao() de cada mensagem.
- **like(indiceMensagem: int)** - aumenta a quantidade de likes da mensagem no índice especificado.

Observação: as classes Mensagem e ListaMensagem são públicas, e as classes MensagemTexto e MensagemImagem tem acesso package. Todos os métodos apresentados devem ter acesso público.

Importante: O programa principal já existe no sistema de correção automática. Submeta apenas as classes especificadas (pacote **chat**). A classes não podem realizar impressão de dados, java.util ou utilizar import.

Casos de teste

Formato dos casos de teste (que aparecem ao avaliar as classes no sistema de correção automática):

Entrada:

- [operação] (índice da mensagem) (dados da mensagem)
 - operação 1: adicionar mensagem de texto
 - operação 2: adicionar mensagem de imagem
 - operação 3: responder com mensagem de texto
 - operação 4: responder com mensagem de imagem
 - operação 9: like
 - operação 10: getMensagens
 - operação 11: getMensagensString

Saída:

- verificação das classes
- métodos executados e saídas obtidas

[VPL](#)

[◀ \[EP\] Conjuntos](#)

Seguir para...

[\[EP\] REC - Dispositivos inteligentes ▶](#)



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

- Conheça a UFABC
- Conheça o NTI
- Conheça o Netel

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos

