INSTITUTO DE COMPUTAÇÃO

Universidade Federal do Rio de Janeiro
Centro de Ciências Matemáticas e da Natureza

Trabalho Prático de Programação Orientada a Objetos

Parte prática do trabalho, que consiste na implementação do sistema proposto, e que poderá ser realizada individualmente ou em dupla. Entregar um relatório individual descrevendo a lógica por trás de cada método implementado e como foi feita a classe Main. Além disso, discorrer sobre um dos tópicos apresentados ao final do enunciado.

Entrega do relatório e dos .java: até 29/06

Novos enunciados (para validação): até 06/06

Implementação:

Dado o diagrama em anexo, implementar o sistema de uma rede social. As classes do sistema estão definidas a seguir:

Recurso

Classe abstrata que tem os seguintes atributos:

- ID, que armazena o ID único do recurso;
- url_recurso, que armazena o caminho que o recurso está disponível.
- prox_ID, que armazena o próximo ID disponível. Inicia com 1.

Esta classe também tem o método abstrato *validaUrlRecurso*, com retorno booleano. Só é possível criar um Recurso com uma url válida.

Video

Classe que herda de Recurso e adiciona os seguintes atributos:

- frame_rate, que armazena a taxa de frames por segundo
- duracao, que armazena o tamanho do recurso em segundos

Esta classe implementa o método validaUrlRecurso comparando se a extensão da URL termina em ‘.mp4’, ‘.mov’ ou ‘.wmv’. Caso seja uma dessas extensões, a URL é válida. Caso contrário, não é válida.

Foto

Classe que herda de Recurso e adiciona o seguinte atributo:

- resolução, que armazena a resolução da foto.

Esta classe implementa o método validaUrlRecurso comparando se a extensão da URL termina em '.jpg', ‘.png' e ‘.bmp’. Caso seja uma dessas extensões, a URL é válida. Caso contrário, não é válida.

Postavel

A interface Postavel é a interface que permite aos diferentes tipos de posts serem vistos como posts. Ela tem os métodos *posta* e *comenta*, ambos sem argumentos que retornem se a operação foi realizada com sucesso ou não.

PostVideo

A classe PostVideo representa uma postagem com um único video. Ela tem os seguintes atributos:

- vídeo que contém um objeto da classe Video
- data_postagem que armazena a data/hora de publicação desta postagem
- lista_comentarios, que armazena todos os comentários feitos nesta postagem
- qtde_fixados, que armazena a quantidade de comentários fixados

É permitido criar um PostVideo sem nenhum vídeo atrelado.

Além desses atributos, a classe PostVideo implementa os seguintes métodos:

- adicionaVideo, que recebe um Video para ser usado na postagem
- posta, que verifica se a postagem tem um vídeo associado e atualiza a data_postagem para a atual caso positivo. Caso não tenha um vídeo associado, a função retorna em estado de erro.

- comenta, que cria um comentário com a data de hoje, a mensagem do usuário e o tamanho da mensagem. Esse comentário criado é armazenado na lista_comentarios do objeto.

PostFoto

A classe PostFoto representa uma postagem com uma sequência de fotos. Ela tem os seguintes atributos:

- qtde_fotos, que armazena quantas fotos compõem a postagem;
- fotos, uma lista que contém os objetos da classe Foto;
- localização, uma String que representa um lugar;
- data_postagem que armazena a data de publicação desta postagem
- lista_comentarios, que armazena todos os comentários feitos nesta postagem
- qtde_fixados, que armazena a quantidade de comentários fixados

É permitido criar um PostFoto sem nenhuma foto atrelada.

Além desses atributos, a classe PostFoto implementa os seguintes métodos:

- adicionaFoto, que recebe uma Foto como parâmetro, adiciona ela na lista fotos e aumenta a qtde_fotos da postagem;
- excluiFoto, que recebe como parâmetro uma Foto e a remove da lista. Caso seja removido, diminui também a qtde_fotos;
- posta, que verifica se a postagem tem pelo menos uma foto e no máximo 10 fotos associadas a ela e atualiza a data_postagem para a atual caso positivo. Caso a validação não dê certo, a função retorna em estado de erro.
- comenta, que cria um comentário com a data de hoje, a mensagem do usuário e o tamanho da mensagem. Esse comentário criado é armazenado na lista_comentarios do objeto.

Comentario

A classe Comentario contém as informações de um comentário através dos seguintes atributos:

- data: que armazena a data que o comentário foi publicado;
- fixado, que representa se o comentário está marcado para aparecer primeiro que os não fixados;

- tamanho, que armazena a quantidade de caracteres no comentário;
- texto. que armazena o texto escrito em si.

PostavelFactory

A classe PostavelFactory tem o método *getPostavel* que recebe uma String e retorna um objeto Postavel dependendo do argumento. Caso o argumento seja “POSTVIDEO”, este método cria um objeto da Classe PostVideo e caso argumento seja “POSTFOTO”, este método cria um objeto da Classe PostFoto. Caso contrário, o método retorna em estado de erro.

Classe de execução (Main):

A classe de execução do trabalho (Main) deverá ter pelo menos os seguintes testes:

- Tentativa de postagem com texto
- Tentativa de postagem com um vídeo atribuído
- Tentativa de postagem sem vídeo
- Tentativa de postagem sem foto
- Tentativa de postagem com 5 fotos atribuídas
- Tentativa de postagem com 11 fotos atribuídas
- Tentativa de criação de comentário em uma postagem com foto
- Tentativa de criação de comentário em uma postagem com vídeo
- Tentativa de criação de vídeo inválido
- Tentativa de criação de foto inválida

Observações para a implementação do trabalho:

- Todos os atributos são privados ou protected. Façam os métodos *getters* e *setters* necessários.
- Os construtores para a maioria das classes foram omitidos. Façam os que acharem necessários.
- Podem adicionar parâmetros aos métodos que acharem necessários, porém, é preciso explicitar essas mudanças no relatório individual.

- Sempre que uma validação não passar, o usuário/desenvolvedor precisa saber o porquê. Pode ser usado Exceções.
- Os pontos da classe Main não precisam ser independentes, só precisam estar assinalados de forma clara no código.
- Ao concluir com sucesso alguma das tentativas da classe Main, imprimir as informações de todos os atributos de cada objeto criado.

Provocação teórica

Dado a implementação deste sistema, discorra sobre pelo menos um dos tópicos a seguir. Cada membro da dupla precisa escolher temas diferentes. Não é necessário explicar de forma específica usando código, basta as ideias usando conceitos de POO. Mínimo de 4 membros, sendo pelo menos 2 deles métodos com explicação.

- Criação do Singleton de Feed
- Criação do Singleton de Configurações
- Classe dos usuários
- Uso de postagens temporárias