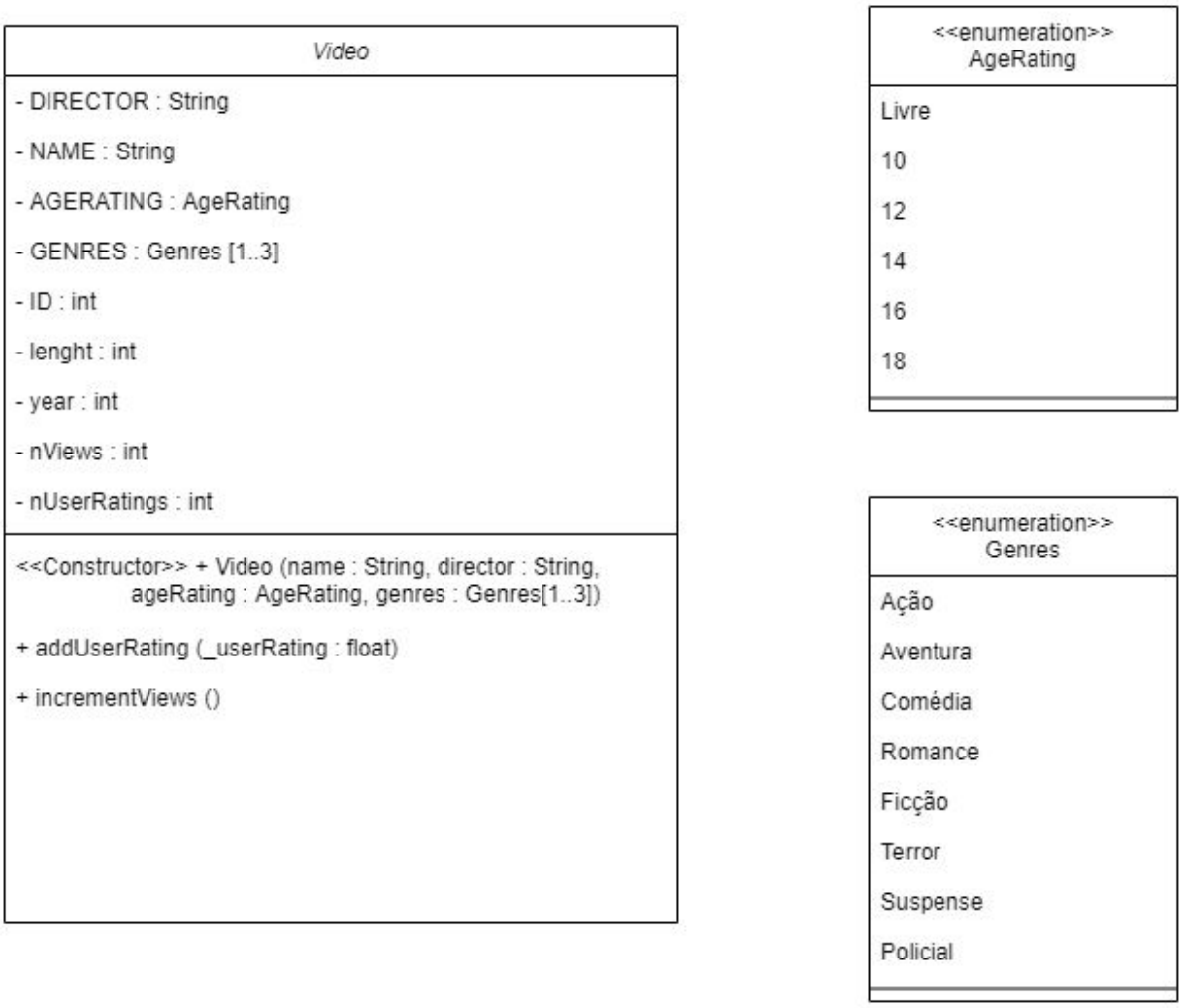
UNIVERSIDADE DE SÃO PAULO - ICMC
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO
SCC0504 - "Programação Orientada a Objetos" - 1ºSem/2020
Exercício 01

Uso de Classes para a construção de uma biblioteca de vídeos

Introdução

- Você deverá usar seus conhecimentos de OO para projetar classes que construam a estrutura básica de uma biblioteca de vídeos.
- Você deve seguir as instruções do trabalho. Crie a(s) classe(s) pedida(s), assim como seus atributos e métodos.
- Lembre-se de criar métodos getters e setters para os métodos privados que não forem *final* (constantes), estes, no caso, não precisam de setters.
- O Diagrama de Classe abaixo representa a classe que deve ser criada, além de dois *enum* (enumeradores), que serão fornecidos para vocês na pasta do exercício no Tidia. Ambos dentro da classe Util.java.



As Variáveis

- A classe vídeo contém diversas variáveis que descrevem os dados importantes do vídeo em questão. Algumas são *final* e outras podem ser alteradas posteriormente à inicialização.
- As variáveis contendo o nome do diretor, nome do vídeo, classificação indicativa, gêneros e identificador são indicadas como constantes no diagrama. Ou seja, só podem ser alteradas no construtor.
    - Portanto, não é preciso criar um Setter para elas, pois não podem ser alteradas.
    - Mas devem possuir Getters.

- As demais variáveis podem ser alteradas após a construção do objeto. Portanto, precisam de Getters e Setters, quando fizer sentido lógico.
- A variável de número de visualizações deve ser inicializada com valor zero e só pode ser alterada pela função *incrementViews()*
- A variável *nUserRatings* deve ser inicializada com 0 e aumentada apenas quando o método *addUserRating()* é invocado
- A variável *userRating* deve ser inicializada com o valor 0, e guardará a média aritmética das avaliações dos usuários. Portanto, cada vez que *addUserRating()* for chamado, você deverá atualizar a média da avaliação do objeto em questão.
- A variável *id* é o identificador único do vídeo. Lembrem-se do que discutimos aula passada sobre variáveis *static* em uma classe e usem isso para construir o identificador :)
- A variável de duração do filme deve ser inicializada com valor *-1* caso nenhum valor seja fornecido
- A variável de ano deve ser inicializada com o valor do ano atual caso um ano não tenha sido fornecido (ver comentário ao final sobre a classe Calendar)

Métodos

- Os métodos *incrementViews()* e *addUserRating()* devem respeitar as condições informadas na seção anterior sobre as variáveis que eles alteram
- O construtor definido no diagrama de classes deve ser respeitado. Crie também um construtor que inicialize TODAS as variáveis.
- Para todas as variáveis não-constantes e que não tenham métodos já definidos de modificação, você deve criar setters. Para todas, crie getters.
- O método para atribuir um ano ao filme deve verificar se o ano é menor que 1878 (ano em que o primeiro filme foi gravado) e, caso seja, imprimir uma mensagem no console: "Não existiam filmes antes de 1878!"

- O mesmo deve ser feito para verificar se o ano do filme é maior que o ano atual + 1, e, caso seja, imprimir a mensagem no console: "O filme não pode ter data de lançamento maior que 1 ano a partir deste!" (ver comentário ao final sobre a classe Calendar)
- O método para retornar a duração deve verificar se o seu valor é igual a *-1*. Caso seja, deve imprimir uma mensagem no console: “Não foi atribuída uma duração para este filme!”, e retornar o valor atual.
- A atribuição de valores para gêneros deve verificar se o *array* passado não é vazio. Caso seja, escrever a mensagem no console: “Não foi passado um gênero!”. E atribuir o valor *null* ao *array*.
- Ela também deve verificar caso existam mais de 3 elementos no *array* de gêneros passado, e imprimir a mensagem: “Mais de 3 gêneros foram passados!” e atribuir apenas os 3 primeiros ao *array* do objeto atual. Use o método *System.arraycopy()* (ver referência sobre *array copy* ao fim se tiver dúvidas).
- Lembre-se de clonar o valor dentro dos arrays caso não encontre nenhuma das duas exceções anteriores. Em Java tudo são referências! Use o método *clone()* (ver referência sobre *array copy* ao fim se tiver dúvidas).
- Lembre-se também de retornar um *clone* do *array* de gêneros na função getter dele. Ou você retornará seu endereço para o usuário, quebrando o encapsulamento!

Lembretes

- Lembre-se de utilizar de getter, setters e construtores para acesso, modificação e inicialização de dados.
- Lembre-se das boas práticas de encapsulamento de dados.
- Comente seu código e nomeie suas variáveis adequadamente.
- Lembre-se das boas práticas de programação:
    - Indentação consistente e legível.

    - Divida seu código em funções sempre que um trecho de código for reutilizado várias vezes ou for semanticamente muito diferente de outro trecho.
    - Java não necessita de uma chamada direta a *delete*, mas é aconselhável atribuir o valor *null* a um objeto antes de descartá-lo da lógica do programa.
- Para verificar o ano atual, usem a Classe GregorianCalendar. Busquem a documentação do java e descubram como usá-la (é fácil, não se preocupem :))

Referências e Extras

- https://dev.to/scottshipp/avoid-getters-and-setters-whenever-possible-c8m
- https://www.javaworld.com/article/2073723/why-getter-and-setter-methods-are-evil.html
- https://softwareengineering.stackexchange.com/questions/21802/when-are-getters-and-setters-justified
- https://developer.ibm.com/tutorials/j-introtojava2/
- https://www.uml-diagrams.org/class-reference.html
- https://pt.wikipedia.org/wiki/Javadoc
- https://www.geeksforgeeks.org/array-copy-in-java/