Uma imagem com sentado, preto, computador

Descrição gerada automaticamente

**Faculdade de Ciências Exatas e de Engenharia**

2019/2020

**Programação Orientada por Objetos**

**Festival de Cinema**

**Docentes:**

Mónica Cameirão

Sergi Bermúdez

**Trabalho realizado por:**

Diego Andrés da Silva Briceño (nº 2043818)

Sílvia da Silva Fernandes (nº 2043118)

Rúben José Gouveia Rodrigues (nº 2046018)

Funchal, 17 de maio de 2020

INDICE

1. **Introdução**

Este relatório tem por objetivo demonstrar e explicar o funcionamento de uma aplicação que permite gerir um festival de cinema, usando para tal a plataforma *NetBeans IDE* e a linguagem de programação *Java*.

Este relatório explicará os objetivos principais deste trabalho, tal como a forma de implementação e os procedimentos realizados para tal fim e a justificação pela qual decidiu-se implementá-los. E no final, é incluído, em anexo, o diagrama UML, que demonstra toda a estruturação do software desenvolvido.

1. **Objetivos**

Este trabalho tem por objetivos, aplicar os conhecimentos adquiridos na unidade curricular Programação Orientada a Objetos com o propósito de criar uma aplicação que permita administrar várias edições de um festival de cinema, podendo o utilizador inserir e consultar a informação que pretende.

Resumidamente, em cada edição do festival, poderão participar vários filmes. Cada filme é caracterizado pelo género, tem um realizador e nele participam vários atores. Um filme não poderá ter mais do que um ator ou atriz principais. Os atores distinguem-se pelo nome e anos de carreira, e só poderão participar no máximo de dois filmes em uma edição. No festival, tanto os filmes quanto os atores competem para ganhar prémios. A cada prémio competem 4 candidatos, cujo vencedor é decidido com base na média das pontuações (1-10) dadas por um conjunto de peritos.

1. **Procedimento e implementação do código**
   1. **Classes**
      1. **Pessoa**

Esta classe, serve para especificar os métodos e atributos básicos que distinguem uma pessoa, ou seja, nome e género. Por esse mesmo motivo, a classe pessoa foi declarada como abstrata (*abstract*), pois apenas idealiza um rascunho. Ao declarar esta classe como abstrata estamos a impedir que se criem instâncias da mesma, o que faz sentido, uma vez que no festival apenas existem atores, realizadores e peritos.

O nome é do tipo *String* e o género do tipo boolean, ou seja, se a pessoa for do género masculino, este atributo será true, e se for do género feminino, false. Sendo assim, no construtor desta classe, definiu-se esses mesmos atributos. Definiu-se dois métodos *getter*: **getNome** e **getGenero**. E também implementou-se os métodos **toString** e **equals**. O **toString** apenas irá imprimir o nome e género da pessoa e o método **equals** serve para comparar se um objeto é igual a outro desta classe.

* + 1. **Edição**

A classe Edição controla toda a informação da respetiva edição do festival.

No início desta classe, efetuou-se uma chamada da *package* “java.util.ArrayList” para podermos utilizar todas as funcionalidades ligadas com ArrayLists.

Esta classe tem como atributos: “numEdicao” e “ano” do tipo int, o ArrayList “filmes” contendo os filmes (classe Filme) que participam na edição, o ArrayList “peritos” onde será guardado os peritos (classe Perito) que irão pontuar os candidatos aos prémios e o ArrayList “premios” que contém todos os prémios (casse Premio) que serão concedidos naquela edição.

O construtor desta classe é declarado como sendo do tipo *protect*, com o propósito de apenas poder ser alterado ou acedido somente por classes da mesma *package*. E apenas recebe como parâmetros o número e o ano da Edição, que serão guardados nas variáveis “numEdicao” e “ano”, respetivamente. Para além disto, inicializa os restantes atributos para, posteriormente, ser inserida a informação nos respetivos Arrays, e também chama o método **inserePremios**.

O método **inserePremios,** tal como o nome indica, insere todos os prémios que os filmes e os atores irão competir. Deste modo, todas as edições terão os mesmos nove prémios em jogo:

* Melhor Ator Principal
* Melhor Atriz Principal
* Melhor Ator Secundário
* Melhor Atriz Secundária
* Melhor Filme
* Melhor Realizador
* Melhor Argumento
* Melhor Cinematografia
* Prémio Carreira

O método **insereFilmes**, insere o filme que o utilizador pretende que participe no festival no ArrayList de filmes.

O método **inserePerito**, insere um determinado perito no ArrayList de peritos da Edição, e também adiciona na matriz de pontuações esse perito, inicializando a zero o ArrayList que contém as pontuações dadas por esse perito.

Nesta classe definiu-se vários métodos *getter*: o método **getPeritos** que retorna a lista com todos os peritos da Edição, o método **getNumEdicao** que retorna o número da Edição, o método **getFilmes** que retorna a lista com todos os filmes que participam na Edição, e por último o método **getPremios** que retorna a lista com o nome de todos os prémios.

Para imprimir os filmes participantes e os prémios, implementou-se, respetivamente, os métodos **imprimeFilmes** (caso ainda não existam filmes registados, imprime uma mensagem com essa informação) e **imprimePremios**. Para além disto, esta classe também possui outros quatro métodos para imprimir informação importante:

O método **listarFilmesMaisPremiados**, tal como o nome indica, imprime os filmes que possuem um ou mais prémios. A classe Filme tem um atributo do tipo int intitulado de “numeroPremios”, caso este atributo seja superior a zero, irá ser imprimido o nome do filme em questão e o número de prémios que este já ganhou.

O método **listarCandidatos** imprime todos os quatro candidatos em todos os prémios. Caso um prémio não tenha nenhum candidato, aparece uma mensagem a dizer que esse prémio não tem candidatos.

O método **listarVencedores** lista os vencedores dos prémios, e o método **listarPontuaçõesOrdenadas** lista os candidatos de cada prémio pela ordem da sua pontuação.

E por fim, o método **toString** irá imprimir o ano e número de Edição e o método **equals** serve para comparar se um objeto é igual a outro desta classe.

* + 1. **Perito**

Os peritos são aqueles que avaliam os candidatos aos prémios, sendo a sua avaliação dada numa escala de 1 a 10.

Esta classe é uma extensão da classe Pessoa, e herda todos os métodos dessa mesma classe. Deste modo, no construtor faz-se uma chamada á superclasse Pessoa para guardar no nome e género os parâmetros recebidos.

Para podermos inserir a pontuação dada por um perito a um dado candidato em um dado prémio, definiu-se o método **inserePontuacao**. Este método recebe como parâmetros: o “premio” (classe Premio) que está sendo avaliado, o parâmetro do tipo int “indiceCandidato”, que corresponde à linha da matriz das pontuações que indica a qual

candidato estamos a atribuir a pontuação, o “indicePerito” (do tipo int) que indica a coluna da matriz das pontuações que indica o perito, e por último o parâmetro “scan” que servirá para o programa ler o que é escrito no teclado. Caso a pontuação inserida esteja entre os limites válidos, é inserido na matriz das pontuações na linha “indiceCandidato” e na coluna “indicePerito” a pontuação.

E por fim, esta classe também possui os métodos **toString** e **equals**. O método **toString** faz uma chamada ao método **toString** da superclasse para imprimir o nome e género do perito e o método **equals** serve para comparar se um objeto é igual a outro desta classe.

* + 1. **Realizador**
    2. **Filme**
    3. **Ator**
    4. **Premio**
    5. **FestivalCinema**
    6. **Main**

1. **Conclusão**
2. **Anexos**