

## **Relatório do trabalho pratico**

*Ruben Santos, Nº41308*

*Gonçalo Domingos, Nº41719*

*Rui Pedro De Almeida Barata, Nº 42872*

*Miguel Garcia Marques, Nº41349*

*07/12/2018*

## Índice

Índice .....	ii
1   Menus.....	1
2   Classes Usadas.....	5
2.1   Espetáculo.....	5
2.2   Artista.....	5
2.3   Musica.....	6
2.4   Parceria .....	6
2.5   Bilhete .....	6
2.6   BilheteVIP.....	7
2.7   EspetaculoCheio .....	7

## 1 Menus

Na Menu nós começamos por inicializar 3 ArrayLists para recolher as listas “armazenadas” nos 3 ficheiros “Espetaculos.dat”, “Artistas.dat” e “Parceiras.dat”. Logo a seguir inicializamos 3 ObjectInputStream, e 3 ObjectOutputStream, para cada um dos ficheiros, para que depois pudéssemos usá-los para escrever e ler dos ficheiros. A seguir fizemos um try and catch para ler as listas que estavam “armazenadas” nos ficheiros para as listas, contando também com duas exceções, a IOException, que é lançada caso haja algum problema com a entrada e saída de dados, e a ClassNotFoundException, que como o nome o diz, é lançada quando não é encontrada uma classe. Depois também temos outro try and catch, mas neste caso este serve para associar os ObjectOutputStream com os FileOutputStream, sendo lançada a exceção IOException.

Logo a seguir abrimos um do (..) while para o primeiro menu.

No primeiro menu damos ao utilizador um menu principal, que tem as opções: Gerenciar Espetáculo, Comprar Bilhete, Gerenciar Artista, Gerenciar Parcerias, Exit. O utilizador só sai do do (...) while se seleccionar a opção Exit. Se seleccionar uma das outras 4 opções vai ser levado a outros 5 menus.

- Se o utilizador clicar na opção Gerenciar Espetáculo vai ser levado a um outro menu, em que é usado outra vez um do (...) while para a escolha das opções, em que só sairá do ciclo se escolher a opção Exit. Neste menu ele terá 5 opções, em que uma delas é o Exit, em que são: Criar Espetáculo, Remover Espetáculo, Mostrar Espetáculo, Estatísticas Espetáculos e Exit.
  - Na primeira opção o utilizador vai criar um novo espetáculo, começando por inserir o nome do espetáculo, o local, a data, a capacidade máxima e os preços dos bilhetes, tanto normais como VIPs. Logo a seguir é lhe perguntado quantos artistas vai ter o espetáculo, e este numero vai ser usado para o ciclo for a seguir onde são inseridos os nomes dos artistas que vão estar presentes, caso o nome do artistas esteja no ficheiro “Artistas.dat” ele vai ser inserido na lista de artistas do espetáculo, caso não esteja é dito que o artista não consta no ficheiro e continua a perguntar outros nomes de artistas. Vai ser perguntado ao utilizador também o número de parcerias que o espetáculo vai ter e este numero vai ser usado para o ciclo for a seguir onde vão ser inseridos os nome das parcerias, se essa parceria existir no ficheiro “Parcerias.dat” vai ser adicionado a lista de parcerias do espetáculo, se não é dito que não consta no ficheiro e continua a perguntar outros nomes de parcerias. Depois é perguntado as horas e os minutos em que começa o espetáculo. Fica o espetáculo criado, depois adicionamos o espetáculo a lista de espetáculos, fazemos clear as listas de artistas e parcerias do espetáculo e escrevemos a lista de espetáculos no ficheiro “Espetaculos.dat”.
  - Na segunda opção o utilizador é levado a um menu em que tem 3 opções: Remover só um Espetáculo, Remover Todos os Espetáculos ou Back.
    - Se o utilizador escolher a primeira opção, é lhe apenas pedido o número de série do espetáculo ( que é uma variável static que esta na

- class Espetáculo que cada vez que é instanciado um objeto espetáculo ela aumenta uma unidade, dando assim um número diferente a cada espetáculo), depois é corrido um ciclo for que “percorre” toda a lista espetáculo a procura do espetáculo que tem o numero de serie inserido, e quando é encontrado é removido.
- Se o utilizador escolher a segunda opção, é lhe pedido outra vez o número de série do espetáculo, e depois é feito apenas clear a lista de espetáculos.
  - Na terceira opção é mostrado outro menu com 6 opções: Mostrar Espetáculos através do Nome, Mostrar Espetáculos através do Local, Mostrar Espetáculos através da Data, Mostrar Espetáculo através do Numero de Criação, Mostrar Todos os Espetáculos e Back.
    - Na primeira opção, como o nome indica, é perguntado ao utilizador qual é o nome do espetáculo que quer ver. Depois de inserido o nome, é corrida a lista de Espetáculos, num for, em que em cada iteração é perguntado se o objeto na posição “i” tem o mesmo nome, se tiver é mostrado esse elemento.
    - Na segunda opção é perguntado ao utilizador qual é o local do espetáculo que quer ver. Depois de inserido o local, é corrida a lista de Espetáculos, num for, em que em cada iteração é perguntado se o objeto na posição “i” tem o mesmo local, se tiver é mostrado esse elemento.
    - Na terceira opção é perguntado ao utilizador qual é a data do espetáculo que quer ver. Depois de inserida a data, é corrida a lista de Espetáculos, num for, em que em cada iteração é perguntado se o objeto na posição “i” tem a mesma data, se tiver é mostrado esse elemento.
    - Na quarta opção é perguntado ao utilizador qual é o Numero de Serie do espetáculo que quer ver. Depois de inserido o Numero de Serie, é corrida a lista de Espetáculos, num for, em que em cada iteração é perguntado se o objeto na posição “i” tem o mesmo Numero de Serie, se tiver é mostrado esse elemento.
    - Na quinta opção é dado print a todos os espetáculos existentes.
  - Na quarta opção, opção das Estatísticas, é primeiramente perguntado ao utilizador o numero de série do espetáculo, e logo a seguir é mostrado um menu com 11 opções: Lucro, Montante Parcerias, Gastos Artistas, Total Bilhetes Vendidos, Total Bilhetes Normais, Total Bilhetes VIPs, Lucro só de Bilhetes, Parcerias Espetáculo, Artistas Espetáculo, Duração Espetáculo e Back. Em todas elas, a lista Espetáculos é percorrida usando um for, e é procurado o espetáculo com o mesmo número de série que o utilizador inseriu, e depois vai se a esse espetáculo e vai se buscar as estatísticas que a opção que o utilizador selecionou.
  - Se o utilizador selecionar a segunda opção, Compra Bilhete, o utilizador vai ter 3 opções: Escolher espetáculo, Listar Espetáculos e Back.
    - Na primeira opção é pedido ao utilizador para inserir o nome do espetáculo, logo de seguida vai aparecer um menu com outras 3 opções: Compra Bilhete, Comprar Bilhete VIP e Back.
      - Na primeira opção é perguntado ao utilizador o numero e bilhetes que quer comprar e para inserir o nome de todas as pessoas que vão comprar bilhete.

- Na segunda opção é exatamente igual, mas para os bilhetes VIPs
  - Na segunda opção são apenas listados todos os espetáculos, para que depois o utilizador possa escolher qual dos espetáculos ele vai querer comprar bilhete.
- Se o utilizador selecionar a terceira opção, Gerenciar Artistas, vai ser levado a um menu com 6 opções: Criar Artistas, Remover Artista, Mostrar Artista, Mostrar musicas de um Artista, Mostrar todos os Artistas e Back.
  - Se for selecionada a primeira opção, Criar Artista, o utilizador terá de inserir o nome do artista, depois é verificado se o artista já existe, se existir é mostrado o artista e acaba esta opção voltando para o menu, senão é pedido para ser inserido as musicas desse artista, os minutos de cada musica e os segundos, e o custo do artista. Depois instancia-se o artista com as informações inseridas e este será inserido na lista de artistas e depois a lista será inserida no ficheiro "Artistas.dat".
  - Se for selecionada a segunda opção, Remover Artista, será perguntado ao utilizador se quer remover só um artista ou se quer remover todos, se quiser remover só um o utilizador terá de inserir o nome do artista e o artista será removido da lista de artistas. Se escolher remover todos, a lista de artistas será esvaziada, usando o clear().
  - Se for selecionada a terceira opção, Mostrar Artista, será pedido ao utilizador que insira o nome do utilizador que ele quer que o programa mostre, este nome vai ser procurado na lista de artistas usando um for, e quando for encontrado, o programa dá print a todas as informações do artista.
  - Se for selecionada a quarta opção, Mostra as musicas de um artista, é pedido outra vez o nome do artista em que o utilizador quer consultar as musicas, e depois este nome é procurado na lista de artistas usando um for, quando encontrado, o programa dá print a todas as musicas desse artista.
  - Se for selecionada a quinta opção, Mostrar Todos os Artistas, é corrida a lista de Artistas usando um for, e é dado print a todos os nomes dos artistas.
- Se o utilizador selecionar a quarta opção, Gerenciar Parcerias, vai ser levado a um menu com 6 opções: Criar Parcerias, Remover Parceria, Mostrar Parceria por Range de preço, Mostrar Todas as Parcerias, Mostrar Parceria com o maior valor fornecido e Back
  - Se for selecionada a primeira opção, Criar Parceria, o utilizador terá de inserir o nome da parceria, depois é verificado se a parceria já existe, se existir é mostrado a parceria e acaba esta opção voltando para o menu, senão é pedido para ser inserido o valor fornecido pela respetiva parceria. Depois instancia-se a parceria com as informações inseridas e esta será inserida na lista de parcerias e depois a lista será inserida no ficheiro "Parcerias.dat".
  - Se for selecionada a segunda opção, Remover Parceria, será perguntado ao utilizador se quer remover só uma parceria ou se quer remover todas, se quiser remover só um o utilizador terá de inserir o nome da parceria e a parceria será removida da lista de parcerias. Se escolher remover todas, a lista de parcerias será esvaziada, usando o clear().
  - Se for selecionada a terceira opção, Mostrar Parceria por range de preço, será pedido ao utilizador um valor mínimo e um valor máximo, depois o programa vai correr todas as parcerias existentes na lista de parcerias usando um for e vai ver quais as parcerias cujo o valor fornecido esteja dentro da range, quando o programa encontrar uma parceria nestas condições será dado print ao nome da parceria.

- Se for selecionada a quarta opção, Mostrar Todos as Parcerias, é corrida a lista de Parcerias usando um for, e é dado print a todos os nomes dos parceiros.
- Se for seleciona a quinta opção, Mostrar Parceira com maior valor fornecido, o programa irá procurar na lista de parcerias, a parceria com o maior valor fornecido, usando um for, quando for encontrado, o programa dará print ao nome da parceria e ao valor que ela fornece.

## 2 Classes Usadas

### 2.1 Espetáculo

A class espetáculo serve para criar diversos espetáculos e tem 22 variáveis de class:

- Data – uma LocalDate que guarda a data do espetáculo;
- numeroSerie – um int estático que vai servir para identificar os espetáculos por ordem de criação;
- nmrSerie – um int que guarda o numeroSerie de um espetáculo;
- Nome – uma string com o nome do espetáculo;
- Local – uma string que contem o local onde se vai realizar o espetáculo;
- nArtistas – um int que contem o número de artistas de um espetáculo;
- capacidade\_maxima – um int com a capacidade máxima de um espetáculo;
- custoBilhete – um double com o custo do bilhete de um espetáculo;
- custoBilheteVIP – um double como custo do bilhete VIP de um espetáculo;
- bilhetesVendidosNormal – um int com o número de bilhetes normais vendidos;
- bilhetesVendidosVIP – um int com o número de bilhetes VIPs vendidos;
- montante – double que contem o dinheiro dos fornecedores;
- horas\_espetaculo – um int que contem a hora de início de um espetáculo;
- minutos\_espetaculo – um int que contem os minutos que um espetáculo inicia;
- durH – um int que contem o tempo, em horas, que um espetaculo demora;
- durM – um int que contem o tempo restante, em minutos, que um espetaculo demora;
- TotalCustoArtistas – um double que contem o custo total de todos os artistas;
- Artistas – um ArrayList com todos os artistas que vão atuar num espetáculo;
- Parcerias – um ArrayList com todas as parcerias dispostas a patrocinar determinado espetaculo;
- Bilhetes – um ArrayList com todos os bilhetes já vendidos de um determinado espetáculo;
- BlhetesVIP – um ArrayList com todos os bilhetes VIPs já vendidos de um espetáculo.

Os métodos presentes são: os *getters*, os *setters*, *duracaoTotal* que calcula a duração que o concerto vai ter tendo em conta todos os artistas, *compraBilheteNormal* e *comprarBilheteVIP* que tratam das compras dos bilhetes normais e VIPs respetivamente, *dinheiroParcerias* que soma a variável montante o dinheiro de todas as parcerias, *custoArtistas* que soma o custo de todos os artistas que vão participar num determinado espetaculo, *lucroEspetaculo* que calcula o lucro que a gerência fez com cada espetaculo, *toString*, *clone* e *equals*.

### 2.2 Artista

A class artista serve para criar os vários artistas que possam atuar num certo espetáculo. Tem 6 variáveis de class:

- nome – uma string que contem o nome do artista;

- `setM` – um `ArrayList` que contem as músicas que um artista vai cantar em palco;
- `horas` – um `int` que contem o tempo, em horas, que um artista vai demorar a cantar as suas músicas;
- `minutos` – um `int` que contem o tempo restante, em minutos, que um artista vai demorar a cantar as suas músicas;
- `segundo` – um `int` que contem o tempo restante, em segundos, que um artista vai demorar a cantar as suas músicas;
- `custo` – um `double` que contem o custo do artista;

Os métodos presentes são: os *getters*, os *setters*, *changeSetM* que muda uma música numa dada posição, *duração* que calcula o tempo de atuação, *toString*, *clone* e *equals*.

## 2.3 Musica

Esta class é usada para criar várias músicas que determinado artista vai cantar em palco. Tem três variáveis de class:

- `nome` – uma `string` com nome de uma música;
- `minutos` – um `int` com tempo, em minutos, de uma música;
- `segundos` – um `int` com tempo restante, em segundos, de uma música.

Os métodos presentes são: os *getters*, os *setters*, *toString*, *equals* e *clone*.

## 2.4 Parceria

A class `Parceria` tem duas variáveis de class:

- `nome` - uma `string` com nome da entidade que combinou a parceria;
- `valor_fornecido` – um `double` com o valor fornecido por determinada entidade.

Os métodos presentes são: os *getters*, os *setters*, *toString*, *equals* e *clone*.

## 2.5 Bilhete

A class `Bilhete` apenas tem duas variáveis de class:

- `custo` – um `double` que define o seu custo, que pode ser diferente de espetáculo para espetáculo;
- `NomeComprador` – uma `string` que contem o nome do cliente que adquiriu tal bilhete.

Os métodos presentes são: os *getters*, os *setters* e o *clone*.



## 2.6 BilheteVIP

A class BilheteVIP é igual a class Bilhete mas neste caso:

- custo - um double que contém o custo do bilhete VIP de determinado espetáculo;
- NomeComprador – uma string que contém o nome do cliente que adquiriu um bilhete VIP.

Os métodos presentes são: os *getters*, os *setters* e o *clone*.

## 2.7 EspetaculoCheio

Esta class é apenas para mandar uma exception se determinado espetáculo já não tem mais bilhetes disponíveis. Como tal não tem variáveis de class.