

FCT Boleia

Relatório de Algoritmos e Estruturas de Dados 2019/20 (MIEI)

Fase 2

Alunos : José Murta (55226) e Diogo Rodrigues (56153)

Curso: Mestrado Integrado em Engenharia Informática

Turno: P7 e P1

Docente prático: Sofia Cavaco

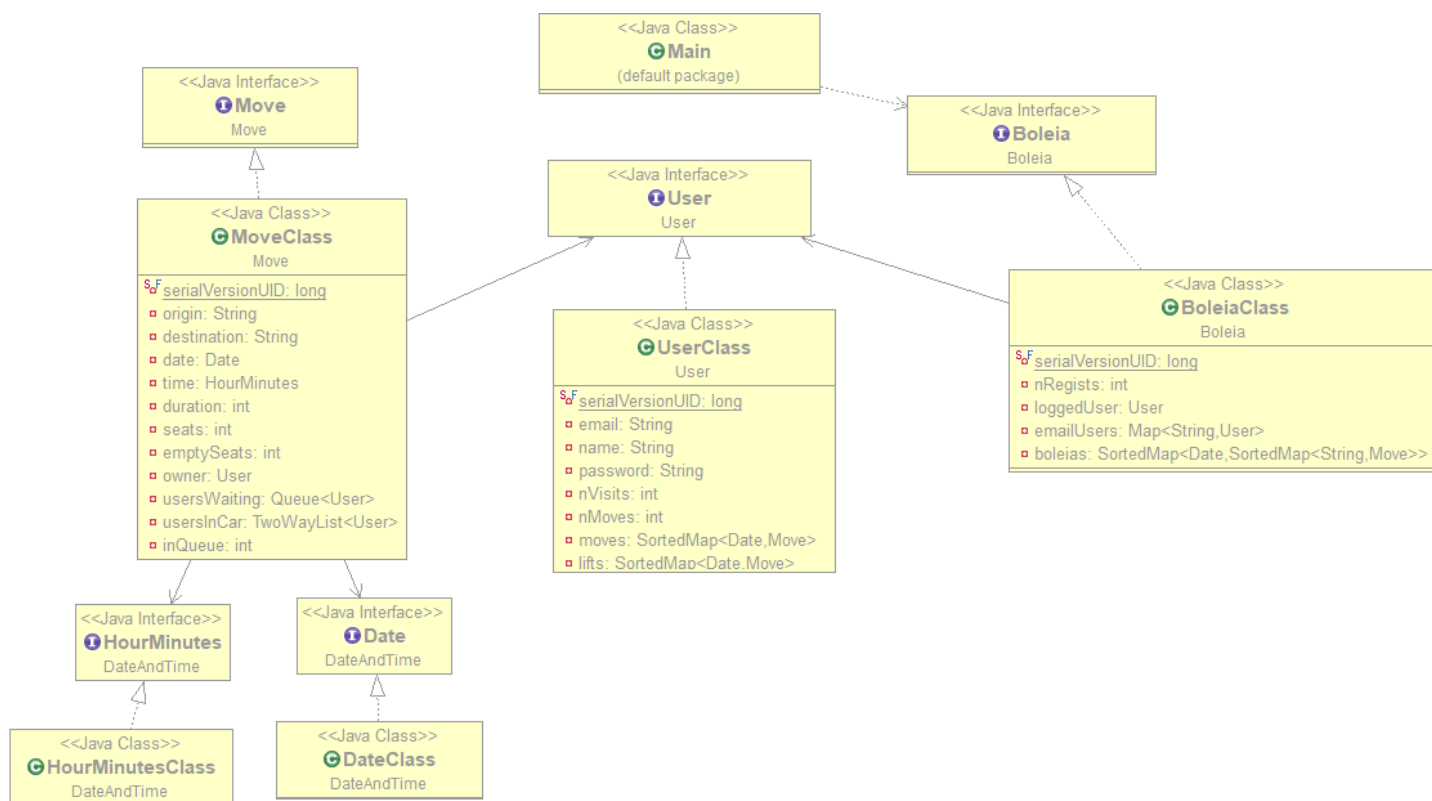


Figura 1 - Diagrama de classes

Descrição e justificação das alterações realizadas fase à 1ª fase do trabalho

1. Criação de um novo objeto – HoursAndMinutes

Inserimos um novo objeto (HoursAndMinutes) de forma a simplificar a leitura da hora. A hora é dada pelo utilizador num formato String HH:MM, e com esta classe dividimos a hora e os minutos, tal como é feito na data, e verificamos se a hora dada pelo utilizador é válida.

2. TwoWayList<Move> moves -> SortedMap<Date,Move> moves

Concluimos que seria necessária a mudança de TAD no objeto User, de lista duplamente ligada para SortedMap, visto as deslocações do utilizador precisarem de estar ordenadas por data.

3. TwoWayList<Move> moves -> SortedMap<Date,Move> moves

Concluimos também que seria necessária a mesma mudança para o tratamento de boleias do objeto User, pois da mesma forma que as deslocações precisam de estar organizadas por data.

4. SortedMap<Date, Move> -> SortedMap<Date, SortedMap<String,Move>>

Devido à necessidade de listagem por data e posteriormente por ordem alfabética do email do utilizador, decidimos mudar a TAD SortedMap de forma a manter no valor um sortedMap em que a chave é uma String (email do utilizador) e o valor a deslocação, cumprindo assim com as necessidades de organização.

Descrição dos comandos do programa

- **Comando Regista:** Cumprindo todas as exceções, é criado um novo objeto do tipo User, e é adicionado o email desse utilizador ao SortedMap emailUsers, sendo a chave a String com o email do utilizador e o valor o próprio objeto User. Através de uma variável User (loggedUser) confirmamos o estado do programa (modo inicial ou sessão).
- **Comando Entrada:** Cumprindo todas as exceções, é realizada a entrada do utilizador caso a password seja compatível, sendo atualizada a variável loggedUser para este utilizador que entrou, e o estado do programa passa para modo sessão.
- **Comando Sai:** Caso o sistema esteja em modo sessão, o utilizador termina a sua sessão, tornando a variável loggedUser null, logo o programa fica em modo inicial.
- **Comando Termina:** Termina a execução do programa, caso o programa esteja em modo inicial (sem nenhum utilizador com sessão iniciada),
- **Comando Ajuda:** Imprime a lista de funcionalidades disponíveis consoante o estado do programa.
- **Comando Nova:** Cumprindo todas as exceções, é adicionada uma nova deslocação com determinadas características ao utilizador com sessão iniciada, adicionando esta deslocação ao SortedMap moves, sendo a chave a data da deslocação e o valor a própria deslocação, e ao SortedMap boleias da classe topo.
- **Comando Remove:** Cumprindo todas as exceções, é removida a deslocação com uma dada data, do utilizador, caso essa deslocação ainda não tenha nenhuma boleia associada e caso não exista mais nenhuma deslocação nessa data é removida a data do SortedMap boleias.
- **Comando Boleia:** Cumprindo todas as exceções enunciadas, é adicionada uma boleia ao utilizador com sessão enunciada, na deslocação criada por um dado utilizador numa dada data, adicionando o ao SortedMap lifts, a data da deslocação como chave e a própria deslocação como valor, e adicionando também o utilizador com sessão iniciada à TwoWayList usersInCar da própria deslocação. Caso essa deslocação esteja sem lugares vazios então o utilizador em sessão iniciada não será adicionado à TwoWayList usersInCar, mas sim à Queue usersWaiting.
- **Comando Retira:** Cumprindo todas as exceções, é removida a boleia, com uma data, ao utilizador com sessão iniciada. Remove do SortedMap lifts, da classe User, essa deslocação e remove o utilizador da TwoWayList usersInCar da própria deslocação, e caso estivesse algum utilizador na Queue usersWaiting, é retirado o primeiro utilizador desta fila e passado para usersInCar.
- **Comando Consulta:** Cumprindo todas as exceções enunciadas, são consultadas as informações da deslocação criada pelo utilizador com email dado, na data dada através da pesquisa Sortedmap boleias, em que a chave é a data, e o value é outro SortedMap em que a chave é o email do utilizador e o value é assim a deslocação que queremos consultar.
- **Comando lista:** Lista Minhas: Lista todas as deslocações do utilizador que se encontra com sessão iniciada. Cumprindo todas as exceções, de forma a listar todas as deslocações iteramos o sortedMap moves no objeto User; Lista Boleias: Lista todas as boleias em que o utilizador com sessão iniciada se encontra inserido. Cumprindo todas as exceções, de forma a listar todas as boleias onde está inserido iteramos o sortedMap lifts no objeto User; Lista DATA: Lista todas as deslocações que existem no programa com a data que foi inserida pelo utilizador. Cumprindo todas as exceções, apenas são imprimidas deslocações com lugares vagos. De forma a listar todas as deslocações iteramos todas as deslocações existentes no sortedMap que é valor do SortedMap boleias na classe topo; Lista EMAIL: Lista todas as deslocações do utilizador com o email que foi dado. Cumprindo todas as exceções, iteramos o sortedMap de moves do utilizador com o email; Lista Todas: Lista todas as deslocações existentes no programa. Cumprindo todas as exceções, iteramos o sortedMap boleias que armazena todas as deslocações do programa.