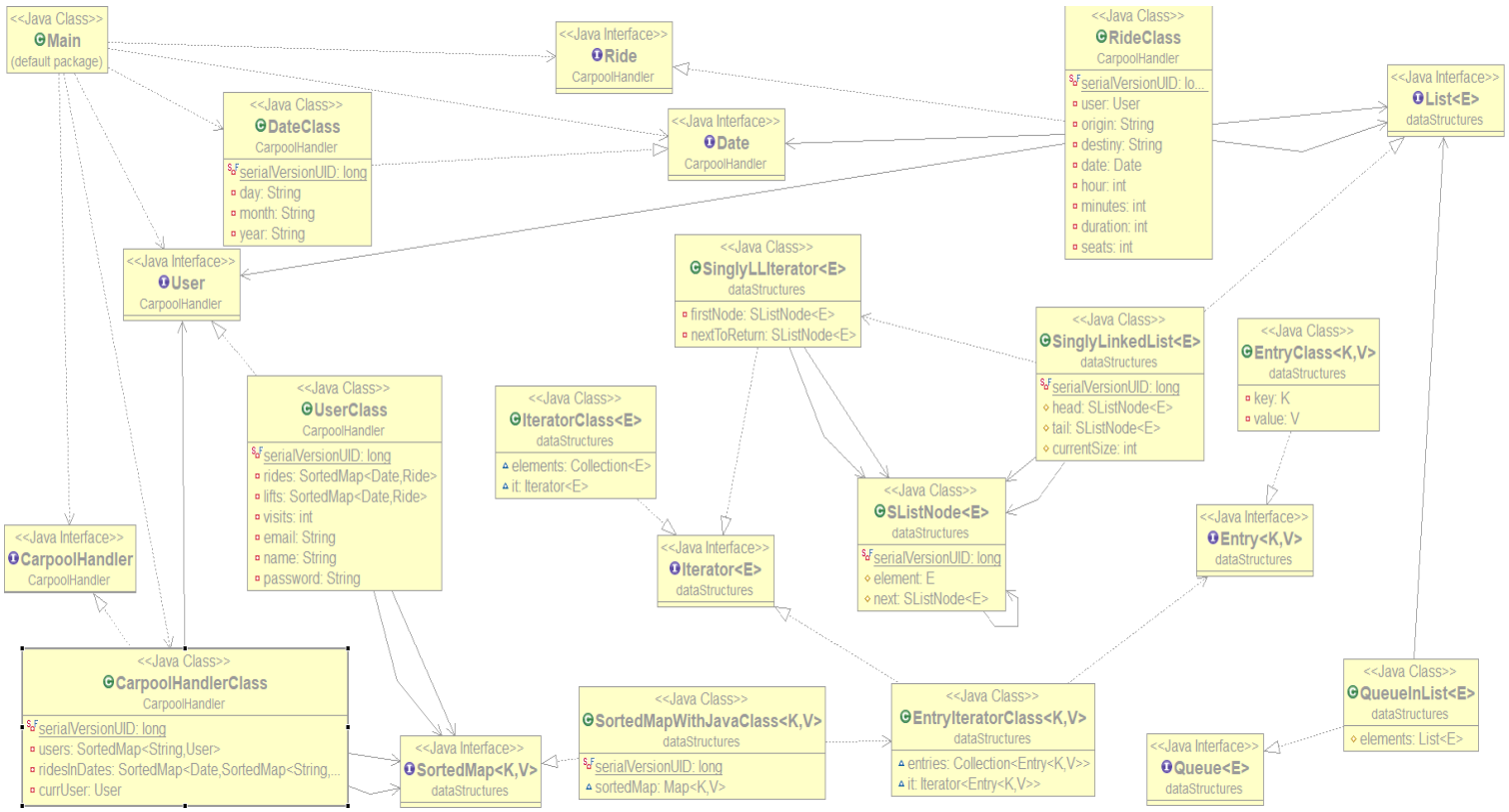

Alterações relativamente à fase 1

Uma das alterações que fizemos foi em vez de na classe CarpoolHandlerClass usarmos um mapa ordenado cuja chave é um objeto Date e o valor uma lista ordenada, usamos um mapa que continua a ter como chave uma Date mas o seu valor é um mapa ordenado(chave corresponde ao email do User que criou a Ride e o seu valor vai ser essa mesma Ride).Fizemos esta alteração visto não termos nenhuma TDA ao nosso dispor que seja uma lista ordenada.

A outra alteração que fizemos foi em vez de termos um mapa ordenado de users em que o email é a chave e o valor seria o objeto User ter um hashMap, visto não haver necessidade de o mapa estar ordenado.

Descrição das funcionalidades/comandos do programa

Comando Lista

O comando Lista vai se desdobrar em quatro subcomandos dependendo da indicação dada, se o utilizador escrever a palavra “minhas” à frente da palavra “Lista” , o comando listaMinhas vai ser executado, se for escrita uma data à frente a listaData vai ser executada, se for a palavra “boleias” a listaBoleias vai ser executada, se for a palavra “todas” a listaTodas vai ser executada e finalmente se for escrito um email de um User a listaEmail vai ser executada.

listaMinhas:

Este subcomando vai listar as deslocações do currUser(variável User dentro da class topo(CarpoolHandlerClass) que guarda o utilizador com a sessão iniciada), para isso o currUser vai devolver um iterator com as suas deslocações e escreve as informações das deslocações.

listaData:

Este subcomando vai listar as deslocações os emails dos Users que têm uma deslocação da Date que é pedida, para isso a classe CarpoolHandlerClass vai devolver um iterador com as deslocações que acontecem nessa Date.

listaBoleias:

Este subcomando vai listar todas as deslocações em que o current user apanha boleia , para isso o currUser devolve à classe topo que por sua vez devolve à Main um iterator com todas as deslocações em que tem boleia e escreve todas as informações da deslocação.

listaTodas:

Este subcomando vai listar todas as deslocações existentes no programa, para isso a CarpoolHandlerClass vai devolver um iterator de Entries do mapa que contem todas as deslocações ordenadas por Date(chave) , que por sua vez lá dentro vai ter um map das deslocações nesse dia ordenado por email do user que tem a deslocação(chave).Este subcomando vai escrever a Date e o email do User da deslocação.

listaEmail:

Este subcomando vai listar todas as deslocações do User cujo email foi dado, para isso a CarpoolHandlerClass vai devolver um iterator com as Rides do User para tal, vai ao mapa de Users que tem para procurar o User e este devolver um iterator com as suas deslocações. Escreve todas as informações das deslocações na consola.

Comando Regista

Este comando vai registar um novo User. Vai ser dado um email, um nome e uma password. O User só pode ser criado se ainda não houver um User com o mesmo email e for dada um password válida. O CarpoolHandlerClass vai guardar o novo objeto User no mapa de users.

Comando Sai

O User que está com a sessão iniciada vai sair da sessão e o programa volta ao modo inicial. Para isso o CarpoolHandlerClass vai igualar o currUser a null e retornar o nome do User que vai sair da sessão. Vai escrever o nome do User na consola.

Comando Entrada

Um User dá entrada no programa com o email e a password. Depois de ser dado um email já existente no sistema, o objeto user vai ser procurado no mapa de users e guardado no currUser.

Comando Termina

Este comando termina o programa, escrevendo a respetiva mensagem de fim de execução.

Comando Nova

Este comando, quando o utilizador fornece os dados necessários e se o utilizador já não tiver uma deslocação ou boleia na data dada, é criado um Objeto Ride com as informações dadas no mapa com todas as deslocações, e no mapa de deslocações do utilizador corrente. Se os argumentos forem inválidos, a data for inválida ou o utilizador já tem uma deslocação ou boleia na data dada, é escrito o output correspondente.

Comando Remove

Este comando, quando o utilizador fornece uma data remove com sucesso a deslocação se este já não tiver utilizadores resgistados na deslocação. A remoção passa por remover a deslocação do mapa de deslocações do utilizador do mapa de todas as deslocações. Se a data não estiver formatada, a deslocação já tiver utilizadores registados ou não existir deslocação nesse dia, é escrito o output correspondente.

Comando Ajuda

Este comando imprime o menu de ajuda dependendo se o programa está em modo inicial ou modo sessão.Para ver isto, é verificado se a variável correspondente ao utilizador corrente está a null(significa que ninguém tem a sessão iniciada) ou se tem algum user lá dentro.

Comando Boleia

Este comando, dado um email e uma data vai tentar adicionar um User a uma Ride, se ainda houver lugar nessa deslocação o user é adicionado à lista de users e a deslocaçao é adicionada ao mapa que o user tem de boleias. Se não houver lugar, o user vai ser adicionado a uma queue que serve como lista de espera para essa deslocação e o programa escreve a posição em que ficou.

Comando Retira

Este comando dado uma data vai retirar do mapa de boleias do user currente a boleia (caso a tenha) e vai retirar da deslocação em si, o user da lista de users.

Comando Consulta

Este comando dado um email e uma data vai procurar no mapa de users o email dado e vai procurar a deslocação correspondente à data dada no map de rides do user e imprime (se essa deslocação existir) todas as informações referentes à deslocação.

David Antunes, 55045

Carolina Duarte, 55645