



Universidade do Minho - Escola de Engenharia
Mestrado Integrado em Engenharia Informática
Processamento de Dados com Streams de JAVA

DateTimeApp

António Jorge Monteiro Chaves A75870
Carlos José Gomes Campos A74745

8 de Dezembro de 2018

Resumo

Este relatório pretende descrever o desenho da implementação da aplicação, desenvolvida no âmbito da unidade curricular de Processamento de Dados com Streams de JAVA. Esta mesma consiste numa calculadora universal de Datas, Tempos e Fusos horários e ainda possui uma agenda. Junto com este relatório, vem também um manual de utilizador para que possa ser consultado em caso de dúvidas.

1 Introdução

1.1 Caso de Estudo

O presente documento consolida todo o processo de desenvolvimento e utilização da aplicação desenvolvida no âmbito da primeira iteração do conjunto de trabalhos práticos da UC de Processamento de dados com Streams de Java.

A aplicação tem como objetivo final a manipulação de objetos da API *Java Time* e suas classes derivadas. Esta está dividida em três partes, uma que é a calculadora de datas e tempos, outra que é a calculadora de fusos horários e por fim a criação e gestão de uma agenda pessoal.

1.2 Estrutura do Relatório

A estrutura deste documento está dividida em 2 partes, sendo que a primeira é o conjunto de justificações para a forma como a aplicação foi codificada. E a segunda é um manual de utilizador, que ilustra a forma de funcionamento do programa.

2 Modelação segundo MVC

O modelo MVC é, especificamente no contexto do desenvolvimento de aplicações de interface de utilizador, um padrão que propõe a divisão da aplicação em três camadas distintas: Modelo(*Model*), Visualização(*View*) e Controlo (*Controller*), de modo a promover uma melhor compreensão do código que incorpora a aplicação, reutilização do mesmo e o seu desenvolvimento paralelo de modo eficiente.

2.1 Model

Nesta camada foram implementadas 4 classes, a *InterfaceDateTimeModel* que representa a interface usada para implementar uma camada de Modelo de dados à nossa escolha. Assim deste modo, se quisermos modificar toda a camada de modelação, basta depois fazer umas pequenas alterações na classe *Main*.

A modelação escolhida pelo grupo de trabalho, está codificada na classe *DateTimeModel*, que tem apenas um *Map* como coleção de dados. Esta decisão foi tomada, para que existisse uma ligação direta entre um dia arbitrário, e o seu conjunto de tarefas correspondentes. Como tal foi escolhido para *Key*, uma *LocalDate*, e para *Value* um objeto da classe *Day*.

Sendo que esta última é uma *List* de *Slot*. Esta escolha foi tomada assim, para que houvesse uma noção de ordem no conjunto de tarefas a executar num dado dia. Por fim a já mencionada classe *Slot*, é constituída por um *LocalTime* para representar a hora de início, outro *LocalTime* para representar a hora de fim, e uma *String* que contém a descrição da tarefa em si.

2.2 View

Nesta camada, tal como na *Model*, também foi implementada uma interface, para que futuramente possa-se fazer uma representação gráfica da aplicação, em vez da textual, fazendo também umas pequenas alterações classe *Main*.

Na *View* existe uma coleção de menus textuais de navegação da aplicação, representados pela classe *Menus*. Esta possui um *Map* como variável de instância, em que as *Keys* são *Integer*, que representam a ordem dos menus ao longo da execução do programa, e os *Values* são objetos instanciados pela classe *Menu*.

Um objeto da classe **Menu** contém um conjunto ordenado de opções, que são manipuladas pelos controladores respetivos, e o seu título.

2.3 Controller

Esta camada foi dividida em várias classes, para que se pudesse obter uma maior modularidade. Esta mesma, engloba o conjunto de todas as operações possíveis da aplicação, tanto de navegação como de queries.

2.3.1 DateTimeInputController

Classe da aplicação cujo intuito é pedir ao utilizador dados relevantes para a criação de objetos da biblioteca *Java Time*, assim como testar a correção das entradas. A título de exemplo, aquando do pedido de elementos constituintes de datas, não só apresenta ordenadamente os elementos a introduzir (dia, mês e ano), como testa a possibilidade de criação de uma data com esses valores, falhando, por exemplo, quando a data inserida é o dia 31 de Fevereiro.

2.3.2 DateTimeFlowController

Controlador desenvolvido com vista a utilizar todas as queries disponíveis sobre calculos de datas e tempos. Faz uso da classe *DateTimeInputController* quando necessita da introdução de elementos do tipo *LocalDate* ou *LocalTime* e executa sobre estes um conjunto pré-definido de operações.

2.3.3 TimeZoneFlowController

Controlador desenvolvido com vista a utilizar todas as queries disponíveis sobre fusos horários. Faz uso da classe *DateTimeInputController* quando necessita da introdução de elementos do tipo *ZonedDateTime* ou *ZoneId* e executa sobre estes um conjunto pré-definido de operações.

2.3.4 ScheduleFlowController

Este sub-controlador possui um conjunto de métodos que representam as funcionalidades *C.R.U.D.* (Create, Retrieve, Update, Delete) do ficheiro de dados que possui a informação da agenda pessoal.

Esta classe faz uso da classe *DateTimeInputController* quando necessita da introdução de elementos do tipo *LocalDate* ou *LocalTime* e executa sobre estes um conjunto pré-definido de operações.

2.4 Main

A classe *Main* possui duas funções que são usadas para ler o ficheiro de dados no início da execução, e para escrever todas as alterações executadas ao longo do programa. O ficheiro de dados tem o nome *Schedule.dat* e está guardado na pasta de projeto.

3 Manual de Utilizador

3.1 Inicialização

Caso apareça algum erro de leitura na consola de execução, é aconselhado a *Sair da Aplicação* através da tecla "Q", e reiniciar a aplicação.

3.2 Operações sobre Datas

O primeiro item do Menu Inicial encaminha para o conjunto de operações definidas sobre datas e tempos.

3.2.1 Diferença entre datas

Após a inserção, por parte do utilizador, da informação relativa a duas datas, retorna a diferença entre elas em dias, meses e anos.

3.2.2 Somar/Subtrair tempo a uma data

Inicialmente é pedido ao utilizador que defina a operação a efetuar. A escolha é definida pelo símbolo "+" para soma ou "-" para a subtração. De seguida, o utilizador deve introduzir os campos da data sobre a qual se vai operar e os valores de dia, mês e ano a utilizar. O método retorna uma data resultante da operação pretendida.

3.2.3 Calculadora de idades

Partindo da introdução de uma data de nascimento, o método calcula a diferença dessa data até ao dia atual, retornando essa diferença, correspondente com a idade da data fornecida.

3.2.4 Diferença entre tempos

Método análogo ao do primeiro ponto, com a distinção do uso de tempos (horas, minutos e segundos) na inserção e retorno.

3.2.5 Somar/Subtrair tempo a um tempo

À semelhança do ponto anterior, o método é congénere com o segundo, diferindo no uso de tempos para efetuar as operações.

3.2.6 Último dia útil do mês

O método executado nesta opção, indica em que dia do ano é que calha o último dia útil de um determinado mês. Uma boa opção para saber quando é a data limite de pagamento de salários.

3.2.7 Dias úteis entre datas

Esta opção permite ao utilizador indicar o número de dias úteis à sua escolha, independentemente da sua ordem cronológica. Esta funcionalidade pode ser utilizada para calcular o número de dias que os funcionários gastam durante o período de Natal ou da Páscoa.

3.2.8 Somar dias úteis a uma data

Por último, neste menu é dado a possibilidade ao utilizador de saber qual é a data correspondente à soma de um número arbitrário de dias úteis. Uma boa maneira de saber qual é a data limite de apresentação num tribunal.

3.3 Operações sobre Fusos

O conjunto de operações sobre fusos pode ser alcançado selecionando a segunda opção do Menu Inicial.

Segue-se o conjunto de operações disponíveis sobre fusos horários. A execução da aplicação fornece, quando necessário, uma lista de catorze fusos horários, demarcados por uma sigla. Note-se que estes são apenas uma forma de facilitar ao utilizador a escolha de um fuso, não estando limitada a essa lista e permitindo, a qualquer momento, a utilização de um dos mais de quinhentos fusos disponíveis, sendo apenas necessário indicar o seu nome exato.

3.3.1 Visualização da Data e Hora para qualquer fuso horário

A primeira das queries sobre fusos é de compreensão direta. É pedida a introdução de um fuso horário e retorna a sua data e hora no instante atual.

3.3.2 Diferença de deslocamentos entre dois fusos

Após a introdução de dois fusos horários, é calculada a sua diferença em horas e minutos.

3.3.3 Cálculo da hora de chegada de uma viagem

A terceira query pretende calcular a hora de chegada de uma eventual viagem, dada a data de partida, número de horas de viagem e fuso horário do local de chegada.

Para a execução correta deste método, é necessário fornecer os três elementos referidos, sendo retornado, no final, a data de chegada no local indicado, tendo em conta o número de horas decorrido.

3.3.4 Cálculo do tempo de uma viagem

Após fornecidos os dados para duas datas, referentes à partida e chegada de uma viagem simulada, é calculada a sua diferença em horas e minutos. A introdução de uma data de chegada que seja anterior à de partida é sinalizada com uma mensagem de erro e a execução da query terminada sem resultado final.

3.3.5 Listagens de fusos horários

É possível pedir qualquer uma de três listagens de fusos horários, total ou apenas por deslocamentos certos ou incertos. As duas últimas referem-se a fusos horários que contenham ou não diferença para UTC+00 em minutos. Um exemplo de um fuso incerto é o do Irão que, do meridiano de Greenwich, está deslocado em três horas e trinta minutos.

3.4 Utilização da Agenda

A ultima opção do Menu Inicial, permite-nos fazer a gestão da nossa agenda pessoal.

3.4.1 Consultar Agenda

Listagem completa e ordenada crescentemente por dia de todas as entradas da agenda. Quando é apresentada a agenda, a consola fica pendurada, para prosseguir com a execução, basta carregar "Enter".

3.4.2 Inserir Tarefa

Para inserir uma tarefa, a aplicação pede que insira o dia em que se vai inserir, a descrição da tarefa em si, a sua hora de início e a hora de fim. Quando é apresentada a tarefa inserida, a consola fica pendurada, para prosseguir com a execução, basta carregar "Enter".

3.4.3 Consultar dia

Para consultar todas as tarefas marcadas para um dia, basta inserir os dados referentes a esse dia, que é devolvido a listagem de tarefas marcadas para esse dia. Quando é apresentada a agenda desse dia, a consola fica pendurada, para prosseguir com a execução, basta carregar "Enter".

3.4.4 Alterar Tarefa

Para alterar uma tarefa é preciso indicar o dia, o índice da tarefa, e a sua nova descrição. Sempre que o utilizador pede para fazer esta alteração, é mostrada a listagem do dia fornecido, para que seja mostrada a listagem das tarefas, e assim o utilizador pode verificar o número da tarefa a alterar, antes de inserir esse índice.

Para alterar as horas de início e/ou de fim é aconselhado a primeiro fazer a remoção da tarefa e depois fazer uma nova inserção, já com os campos novos.

3.4.5 Remover Tarefa

Para remover uma tarefa é preciso indicar o dia e o índice da tarefa. Sempre que o utilizador pede para fazer esta remoção, é mostrada a listagem do dia fornecido, para que seja mostrada a listagem das tarefas, e assim o utilizador pode verificar o número da tarefa a remover, antes de inserir esse índice.

4 Conclusão

O grupo pensou em várias formas de implementar este projeto, e foram tomadas várias estratégias e decisões que iremos justificar agora na Conclusão.

Houve uma discussão sobre se o *Map* usado na classe *DateTimeModel* deveria ser um *HashMap* ou *TreeMap*. Escolhemos a primeira opção, porque só precisamos de ordem das chaves quando fazemos a listagem da agenda completa, e essa listagem é feita por um stream que percorre as chaves por ordem cronológica, para que seja feita depois a sua impressão, e assim recorremos ao *HashMap* por motivos de eficiência.

Uma futura alteração a fazer neste projeto, é a implementação da *View* em modo gráfico, com ajuda de por exemplo, da biblioteca *Swing*.

Fazemos com o utilizador insira o índice da tarefa nas opções de remover e alterar, porque a ideia era para quando fazer a alteração para uma *View* gráfica, fazer tanto a alteração como a remoção na própria janela de consultar um dia.

Este grupo de trabalho sente que concluiu este trabalho com sucesso, e com ele aprendeu a dominar a metodologia *MVC* e o uso da biblioteca *Time*.