

Universal Date and Time Calculator (UDTC)

Manual de utilização

Afonso Silva a70387

Octávio Maia a71369

Manual de utilização do conjunto de ferramentas de manipulação de datas e tempos escrita com as novas funcionalidades do JAVA8.

Conteúdo

Introdução	1
Utilização	1
Calculadora de idades	3
Calendário	6
Cronómetro	6
Calculadora de datas	6
Agenda	6
Calculadora de tempos	10
Viagens	10
Relógio universal	11
Implementação	11
Arquitectura	11
Controlo de versões	12
Documentação	13
Conclusão	13

Introdução

Utilização

Obtenção do programa

O executável deste programa pode ser descarregado a partir desta ligação:

<https://github.com/ajcerejeira/udtc/raw/master/udtc.jar>

Também pode descarregar o todo o conteúdo do projeto através do `git`:

```
git clone https://github.com/ajcerejeira/udtc
cd udtc/
```

Execução e compilação

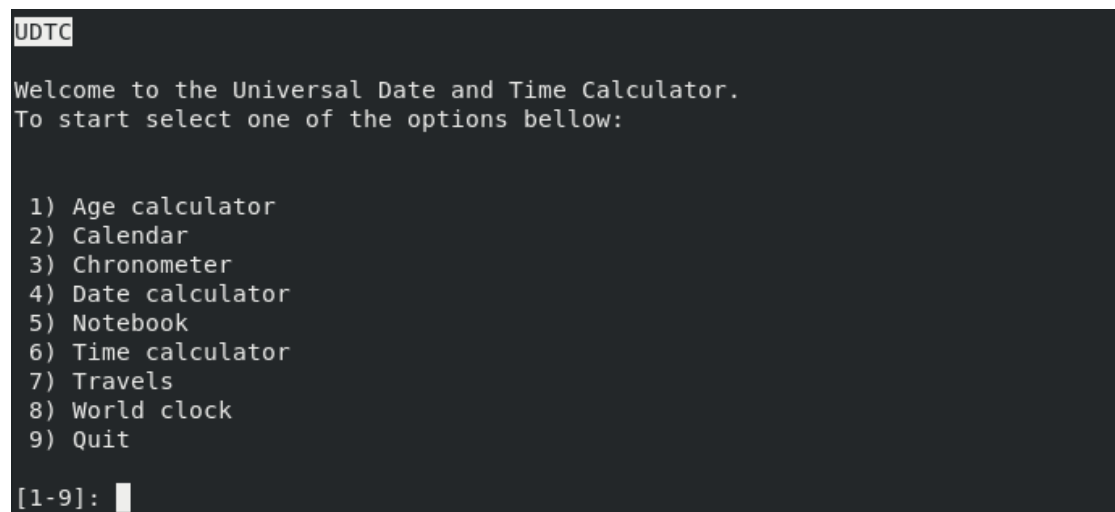
Na pasta raiz do projecto encontra-se o ficheiro `udtc.jar` pronto a ser executado. Deste modo, para iniciar o programa basta correr o seguinte comando na raiz do projeto:

```
$ java -jar udtc.jar
```

Pode também compilar o programa se assim o desejar. O seu código-fonte encontra-se na pasta `src` e pode ser compilado recorrendo à ferramenta `javac` ou então, através de um ambiente de desenvolvimento integrado que suporte `JAVA`.

Após o programa se encontrar compilado, e encontrando-se na directoria onde se encontram os resultados da compilação, basta correr o seguinte comando para iniciar a aplicação:

```
$ java Main
```



```
UDTC
Welcome to the Universal Date and Time Calculator.
To start select one of the options bellow:

1) Age calculator
2) Calendar
3) Chronometer
4) Date calculator
5) Notebook
6) Time calculator
7) Travels
8) World clock
9) Quit

[1-9]:
```

Figura 1: Menu inicial do programa

Navegação pela interface

Tal como sugerido no enunciado deste trabalho prático, o modo de utilização da aplicação passa pela navegação de menus puramente textuais. Cada um destes menus apresenta

um título (e alguns também um subtítulo) bastante visíveis, indicando o modo atual em utilização da calculadora. De seguida é apresentada uma lista numerada com todas as opções disponíveis no modo atual. O utilizador deve então introduzir a opção pretendida no *prompt* e pressionar a tecla <ENTER>, sendo conduzido para o menu seguinte.

Prompt

O *prompt* (pequeno texto que aparece antes do local de introdução do utilizador) indica para cada menu qual o intervalo de opções disponíveis. Regra geral a última opção é a de saída ou retorno ao menu anterior. Quando o utilizador introduz um número inválido, o programa pede para repetir a sua escolha novamente.

```
[1-9]: 12
The chosen option is not valid. Please try again.
[1-9]: █
```

Figura 2: Aviso de introdução de opção incorrecta

Calculadora de idades

```
Age calculator

Time until X years old

Birthday date: [yyyy-mm-dd]: 1996-10-05
How old do you want to be?: 52
You'll be 52 years old in 30 years 9 months and 0 days

1) Back

[1-1]: █
```

Figura 3:

A calculadora de idades permite efetuar algumas operações relacionadas com datas de aniversário, permitindo calcular uma determinada idade, quanto tempo até ao próximo aniversário e quanto tempo demorará a atingir uma determinada idade.

```
Age calculator

Age

Birthday [yyyy-mm-dd]: 1996-10-05
You are 21 years old.

1) Back

[1-1]: 
```

Figura 4: Menu do cálculo da idade

```
Age calculator

Time until birthday

Birthday [yyyy-mm-dd]: 1996-10-05
Time until your next birthday:
0 year(s), 9 month(s), 0 day(s)

1) Back

[1-1]: 
```

Figura 5: Menu do cálculo do tempo até ao próximo aniversário

```
Age calculator

Time until birthday

Birthday [yyyy-mm-dd]: 2018-01-05
Today is your birthday! Happy birthday!

1) Back

[1-1]: 
```

Figura 6: Quando a data de aniversário corresponde ao dia de hoje, o utilizador recebe uma mensagem de parabéns

```
Age calculator

Time until X years old

Birthday date: [yyyy-mm-dd]: 1996-10-05
How old do you want to be?: 52
You'll be 52 years old in 30 years 9 months and 0 days

1) Back

[1-1]: █
```

Figura 7: Menu do cálculo do tempo até se atingir uma determinada idade

```
Calendar

1) Show calendar

[1-1]: █
```

Figura 8: Menu inicial do calendário

```
Chronometer

1) Start
2) Back

[1-2]: █
```

Figura 9: Menu inicial do cronómetro

Calendário

Cronômetro

O cronômetro permite contar o tempo desde a sua execução, permitindo efetuar pausas e retomar a contagem. Os resultados da contagem são mostrados sempre que o cronômetro é pausado ou parado. Infelizmente, por se tratar de uma interface textual é bastante complicado imprimir o valor da contagem em tempo real, pois tal envolveria mecanismos de limpeza e escrita de ecrã que não são bem suportados em muitas plataformas. Em todo o caso, a interface mantém-se simples e de fácil utilização.

```
Chronometer
Chronometer is running

1) Pause
2) Stop

[1-2]: 
```

Figura 10: Menu de início da execução do cronômetro

```
Chronometer
Chronometer has been paused...
Elapsed time: 0:00:09

1) Resume
2) Stop

[1-2]: 
```

Figura 11: Menu de pausa do cronômetro

Calculadora de datas

A calculadora de datas é um dos modos mais extensos e mais completos deste programa.

Agenda

A agenda consiste na agregação de diversos eventos, que são simplesmente compostos por uma data de ocorrência e uma breve descrição. Para inserir e remover eventos utilizam-se

```
Chronometer

Chronometer has stopped...

Elapsed time: 0:00:30

1) Back

[1-1]:
```

Figura 12: Menu de paragem do cronómetro

```
Date calculator

1) Add to date
2) Subtract from date
3) Difference between two dates
4) Week number and day of a date
5) Weekend list
6) Weekdays list
7) Back

[1-7]:
```

Figura 13: Menu inicial da calculadora de datas

```
Notebook

1) Add appointment
2) Delete appointment
3) Search appointment
4) Show appointments
5) Save notebook
6) Read notebook
7) Back

[1-7]:
```

Figura 14: Menu inicial da agenda

as opções 1) Add appointment e 2) Delete appointment respetivamente.

```
Notebook

Add appointment

Date [yyyy-mm-dd hh:mm]: 2018-01-23 15:00
Appointment: Jogo de futebol
```

Figura 15: Menu de inserção de eventos

```
Notebook

Delete appointment

1) [2018-01-03 02:00] Aula de dúvidas de CSI
2) [2018-01-04 02:00] Teste de CSI
3) [2018-01-05 12:00] Entrega de PSDJ
4) [2018-01-08 08:00] Aniversário do pai
5) [2018-01-11 12:00] Teste de EM
6) [2018-01-23 03:00] Jogo de futebol

[1-6]: 6
```

Figura 16: Menu de remoção de eventos

Obviamente que é possível visualizar todos os eventos da agenda, como também filtrá-los por data e hora. Nestas listagens os eventos aparecem sempre ordenados por data

Serialização

É possível guardar e carregar o estado de uma agenda para um formato textual. Para tal utilizam-se os comandos 4) **Save notebook** para guardar num ficheiro os eventos que se encontram na agenda, e 5) **Read notebook** para ler de um ficheiro esses mesmos eventos.

Abaixo se ilustra um exemplo de um ficheiro de dados gerado pela agenda:

```
[2018-01-03 02:00] Aula de dúvidas de CSI
[2018-01-04 02:00] Teste de CSI
[2018-01-05 12:00] Entrega de PSDJ
[2018-01-08 08:00] Aniversário do pai
[2018-01-11 12:00] Teste de EM
```

Ao extrair a agenda para um formato textual torna-se então possível ao utilizador editar a própria agenda mesmo fora da aplicação. Para tal basta utilizar um editor de texto para


```
Notebook

View appointments.

[2018-01-03 02:00] Aula de dúvidas de CSI
[2018-01-04 02:00] Teste de CSI
[2018-01-05 12:00] Entrega de PSDJ
[2018-01-08 08:00] Aniversário do pai
[2018-01-11 12:00] Teste de EM
[2018-01-23 03:00] Jogo de futebol

1) Back

[1-1]: █
```

Figura 17: Visualização de todos os eventos

```
Notebook

Search appointments

From: 2018-01-04 00:00
To: 2018-01-08 23:00
Appointments between 2018-01-04T00:00 and 2018-01-08T23:00
[2018-01-04 02:00] Teste de CSI
[2018-01-05 12:00] Entrega de PSDJ
[2018-01-08 08:00] Aniversário do pai

1) Back

[1-1]: █
```

Figura 18: Filtragem de eventos consoante a data

inserir ou remover novos eventos neste ficheiro, e posteriormente relê-lo com a aplicação. Esta funcionalidade também permite a fácil partilha de agendas entre vários utilizadores.

Calculadora de tempos

Viagens

```
Travels

1) Show all travels
2) Add travel
3) Remove travel
4) Travels between dates
5) Next travel
6) Last travel
7) Check time at arrival
8) Cheapest travels
9) Most expensive travels
10) Shortest travels
11) Longest travels
12) Save travels
13) Read travels
14) Back

[1-14]:
```

Figura 19: Menu inicial das viagens

O modo de gestão de viagens inclui várias ferramentas de conveniência para o cálculo de estimativas de duração de viagens, cálculo de tempos de chegada,

Serialização

Tal como a agenda, o modo de gestão de viagens também permite guardar e ler a lista das viagens para um formato textual.

Exemplo de ficheiro de viagens:

```
[2017-08-01 01:00] Europe/Lisbon - Africa/Kampala (9:30) 480.0€
[2017-09-01 09:00] Africa/Kampala - Europe/Lisbon (10:00) 400.0€
[2018-01-20 13:00] Europe/Lisbon - Europe/London (01:00) 50.0€
```

```
World clock

1) Difference between two time zones
2) Current time at zone
3) World clock
4) Zones at time
5) Back

[1-5]:
```

Figura 20: Menu inicial do relógio universal

Relógio universal

Implementação

Arquitetura

Para garantir uma separação de componentes baseada na sua função, recorreu-se ao padrão de arquitetura de *software* [MVC](#).

Deste modo o projeto dividiu-se nos três *packages* que compõem a arquitetura MVC:

- **Model** - núcleo do projeto, definia interfaces e suas implementações para os diversos modos da aplicação. Neste projeto insistiu-se numa primeira fase na definição de **Interfaces** de **JAVA** que ditavam os métodos que determinado modo deveria seguir. Aliás, garantiu-se que os outros *packages* só acediam a métodos definidos nas interfaces. Estando as interfaces já definidas começou-se a codificação das classes que as implementavam.
- **View** - que para cada interface definida no *package Model* codificava a interação textual com o utilizador.
- **Controller** - que instanciava objetos que implementavam as interfaces definidas no *package Model* e os passava para a **View**.

Com esta arquitectura o processo de desenvolvimento do projeto decorreu com sucesso. Reforça-se ainda a ideia bem sucedida de definir primeiro as interfaces e só depois as implementar, o que facilitou a divisão das tarefas: era assim possível implementar as *view* para essa interface ao mesmo tempo que a sua implementação.

Foi ainda criado um outro *package* denominado **Utils** que continha código utilizado por todos os outros *packages* de utilidade geral. Este era composto pelas seguintes classes:

- **UI** - classe composta por métodos estáticos que facilitam a renderização da interface textual. Inclui o desenho de títulos, subtítulos, menus e introdução textual do utilizador, e impressão de listas de objetos. Com recurso às novas funcionalidades

do JAVA 8 (lambdas, referências a métodos, `Optional`) é muito mais simples e eficaz a criação de novas interfaces.

- `Option` - classe auxiliar utilizada simplesmente para o desenho de menus. Estes são compostos por várias `Option`, sendo que cada uma contém um determinado título e a ação a executar quando esta opção é escolhida. Esta ação não é mais do que um objecto que implemente a interface `Runnable`.
- `Parsers` - classe auxiliar que inclui *parsers* de datas, tempos, durações, zonas e números. Cada parser apresenta a seguinte assinatura: `static Optional<T> parse(String input)`, onde T é o tipo de dados que se pretende obter com o resultado do *parsing*. Repare-se na utilização da nova classe do JAVA 8, `Optional` para indicar se o *parsing* foi ou não bem sucedido.

Controlo de versões

Tal como ditam as regras do bom desenvolvimento de *software*, utilizou-se a ferramenta `git` como método de controlo de versões durante o desenvolvimento do programa. Esta ferramenta foi bastante útil para a coordenação do projeto entre os dois membros do grupo: cada um utilizou um **branch** próprio onde ia desenvolvendo as suas soluções para os problemas, evitando assim conflitos no código. No final os **branches** foram agregados resultando num projeto sustentável e modular.

Todo o código do projeto (este relatório inclusive) foi hospedado no GitHub no seguinte repositório:

<https://github.com/ajcerejeira/udtc>

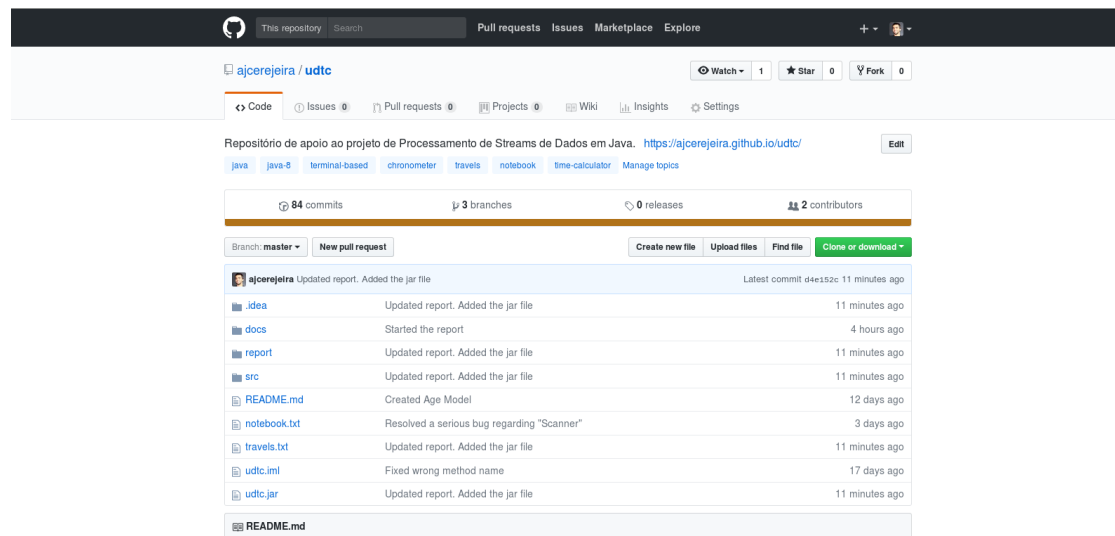


Figura 21: Repositório do projeto no GitHub

Documentação

Toda a documentação do código fonte foi efetuada recorrendo à ferramenta JAVADOC e hospedada como um site do GitHub pages. Esta pode ser consultada diretamente em <https://ajcerejeira.github.io/udtc/>. Lá encontram-se documentadas as classes criadas propositadamente para este programa, encontrando-se informações sobre o seu propósito, implementação e modo de utilização.

The screenshot shows the Javadoc documentation for the `Interface IChronometer`. The left sidebar lists the project's package structure, including `Controller`, `Model`, `Utils`, and `View`. The main content area displays the `Model` package and the `Interface IChronometer`. It includes the `public interface IChronometer` declaration, a description of the interface's purpose, and a code example showing how to use the interface. The `Method Summary` table lists the `pause()` method.

Overview: PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY NESTED FIELD CONSTR METHOD DETAIL FIELD CONSTR METHOD

Model

Interface IChronometer

All Known Implementing Classes:

Chronometer

public interface **IChronometer**

This interface sets the core functionality of a simple chronometer. It has only four possible actions: start, stop, pause and resume. This interface is abstract enough so that its implementations can vary in design. One simple example of its usage could be like this:

```
public static void run(IChronometer chronometer) {
    chronometer.start();
    System.out.println("Start");

    Duration d = chronometer.stop();
    System.out.println("Stop. Elapsed time:" + d);
}
```

Author:

Octavio Maia

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method and Description	
java.time.Duration	pause()	Pause the chronometer and return the time the chronometer has been running.

Figura 22: Exemplo de página da documentação do projeto

Conclusão