

Ana Isabel Sampaio (AIS)

ESINF REPORT



**October 2022**

**2DN**

1210817, João Rodrigues

1210821, Mateus Fernandes

1180727, Ruben Martins

1200882, Gonçalo Teixeira

**INDÍCE**

[1. Diagrama de Classes 1](#_Toc116303297)

[2. Funcionalidades 2](#_Toc116303298)

[**2.1. Exercise 1** 2](#_Toc116303299)

[**2.2. Exercise 2** 2](#_Toc116303300)

[**2.3. Exercise 3** 2](#_Toc116303301)

[**2.4. Exercise 4** 4](#_Toc116303302)

[**2.5. Exercise 5** 6](#_Toc116303303)

[3. Possíveis Melhoramentos 8](#_Toc116303304)

**Índice de figuras**

[Figura 1- Método minCountries 3](#_Toc116303152)

# **Diagrama de Classes**

# **Funcionalidades**

## **2.1. Exercise 1**

## **2.2. Exercise 2**

## **2.3. Exercise 3**

A funcionalidade referente ao exercício 3 permite-nos descobrir o número de países que, em conjunto consegue ter uma quantidade superior a uma certa quantidade fornecida pelo utilizador. Para tal, criamos o método minCountries, que recebe como parâmetros um long quantidade e um TreeMap com String como Key e um TreeSet de Year como Value, e retorna um int.

Começamos por criar uma variável int count, que será o nosso Contador a ser retornado, uma variável long minQuant, variável essa que será posteriormente utilizada para gerir os cálculos das quantidades dos países. Foi criado também um mapa que irá guardar os nomes dos países como Key e a quantidade total de produção de cada país como Value.

Após a criação das variáveis necessárias para a execução do método, foi necessário percorrer o mapa com os dados carregados do ficheiro CSV. Para isso, foram utilizados dois For Each, um a percorrer cada entry do mapa e outro a percorrer cada Set de years. Dentro do primeiro ciclo for os nomes dos países e o total da quantidade para cada país, são adicionados no mapa criado anteriormente, o totalValue. Logo de seguida, esse mesmo mapa foi ordenado por ordem descendente dos seus valores de quantidade.

Após termos o nosso mapa com os valores pretendidos, basta apenas percorrer o mapa e verificar o número mínimo de de países a ser devolvido pela função. Percorremos o nosso mapa sorted e verificamos se o 1º valor é maior que a quantidade, se sim o nosso count fica igual a 1 e esse valor e devolvido. Caso isso não aconteça, o valor é adicionado á variável minQuant criada anteriormente e o valor do count incrementado. Este ciclo é repetido até o valor do minQuant ser maior que a quantidade presente como parâmetro no método.

Uma imagem com texto

Descrição gerada automaticamente

**Figura 1- Método minCountries**

## **2.4. Exercise 4**

O exercício 4 pede-nos para criar um método que ao receber um fruto F, cria uma estrutura adequada, com o número de anos consecutivos em que houve um crescimento da produção desse fruto F e com os países agrupados por ano consecutivos.

Para resolver este exercício optamos por criar um mapa, em que as chaves seriam os anos consecutivos, como inteiros e os valores seriam os Set’s com os países agrupados pelos anos consecutivos de aumento de produção.

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

Foi criado o método “crescimentoConsecutivo” que recebe por parâmetros o fruto F (Item fruto) e recebe o mapa que foi criado anteriormente com os dados do ficheiro CSV. Foi criado inicialmente um TreeMap que vai ser o mapa que o método irá retornar e após isso um objeto item e um objeto “Value”. Primeiro, começamos por ter um ciclo for que percorre todos os países e dentro de esse for é criado um objeto do tipo “Year“e duas variáveis, uma “temp” que servirá de suporte para contar os anos consecutivos e uma “tempmax” que será os anos consecutivos contados que irão ser adicionados ao mapa. Antes de passar ao próximo ciclo for, é criado um Set que é onde vão ser inseridos os países para ser inseridos no mapa que irá ser retornado no fim pelo método.

Após isso, entramos em outro ciclo for que vai percorrer os anos de produção do país que estamos a verificar e dentro desse ciclo temos um if que verifica se o fruto que é produzido no ano que nos encontramos a verificar é o mesmo que recebemos por parâmetro no método. Se a condição for falsa, voltamos ao ciclo for e vamos iterar para o próximo ano, mas se a condição for verdadeira, é verificado de seguida por um if, se a quantidade produzida no ano n+1 é superior à quantidade produzida em n, sendo n, o ano em que nos encontramos no ciclo for e n+1, o consequente a esse mesmo ano. Se a condição for verdadeira incrementamos à variável “temp“ uma unidade, caso contrário, é lhe colocado o valor a 0, e depois é verificado se a variável “temp” é superior a “tempmax”, e se esta condição for verdadeira então atribui-se a “tempMax” o valor de “temp”. No fim, é atribuído a “yearsupport” o valor de “year”, para se poder fazer a verificação n e n+1, já que o for each não permite pesquisar diretamente o valor do ano consequente ao que nos encontramos. Finalmente, após termos percorrido todos os anos de um certo país, iremos adicionar os valores ao mapa. Temos, portanto, uma condição if que verifica se o mapa não está vazio, sendo que se esta for verdadeira, é feita outra verificação no qual pesquisamos se o mapa contém uma key igual ao valor de “tempmax”, e que caso seja verdadeira, adicionamos ao set relacionado a essa key, o país que acabamos de percorrer os anos, caso contrário, é adicionado a um set vazio o país e colocado no mapa numa posição nova, a key “tempmax” e o set com os países associados. Caso o mapa, esteja vazio, adicionamos a key e o set numa posição nova.

Mais abaixo, temos outro método, imprimirMapa, que imprime o mapa acima criado. Ele recebe como parâmetros um mapa com os mesmos atributos do que foi criado no método anterior. Dentro do método, temos um ciclo for que percorre o conteúdo do mapa e imprime tanto a key como as strings dentro dos Set’s.

## **2.5. Exercise 5**

# **Possíveis Melhoramentos**