Съдържание

[Документация по кода 2](#_Toc196596971)

[Файлова структура 2](#_Toc196596972)

[Задача 1: 3](#_Toc196596973)

[Задача 2: 5](#_Toc196596974)

[Задача 3: 7](#_Toc196596975)

[Задача 4: 8](#_Toc196596976)

[Задача 5: 9](#_Toc196596977)

# Документация по кода

## Файлова структура

A screenshot of a computer

AI-generated content may be incorrect.

(фиг. 1)

Всяка задача е разделена в отделен package. Имам изглед в конзолата (Menu). В mapper package, имаме класове, които четат CSV и съответно JSON. Първият го ползваме в задачите, които изискват съхраняването и четенето от Excel. Вторият съдържа примерни низове (думи, специални знаци и емоджита).

## Задача 1:

A computer screen shot of a program code

AI-generated content may be incorrect.

(фиг. 2)

DataWrapper е помощен клас, който се използва да съхранява по горе описаните низове. EXAMPLE съдържа стойността на getWords().

A screen shot of a computer program

AI-generated content may be incorrect.(фиг. 3)

A computer screen shot of a computer code

AI-generated content may be incorrect.

(фиг. 4)

В първа задача правим проверка дали не се съдържа стойност == null или ако низът е празен. Използваме codePoints, заради емоджитата. Понеже не може да се ползва StringBuilder/StringBuilder съм използва for loop, който итерира в обратен ред. Използва конкатенация, за да добави всяка следваща дума.

A black background with white and blue text

AI-generated content may be incorrect.

(фиг. 5)

От менюто избираме опция 1. Първо се визуализира низа, който искаме да обърнем и съответно резултата.

A screen shot of a computer

AI-generated content may be incorrect.

(фиг. 6)

При null или empty се визуализира това.

## Задача 2:

A screen shot of a computer program

AI-generated content may be incorrect.

(фиг. 7)

REGEX\_PATTERN се използва както е по условие. REGEX\_SPLIT съм го отделил, за да не директно в .split(). Проверка дали е null или empty. Ползвал съм iterator, за да мога да премахвам, докато итерирам. Всяка дума, която не отговаря на regex се премахва. Ако hashMap съдържа тази дума само я инкрементираме. Иначе добавяне нови думи в map-a. По-късно сортираме първо по повтаряния и по азбука.

A screenshot of a computer

AI-generated content may be incorrect.

(фиг. 8)

A screen shot of a computer

AI-generated content may be incorrect.

(фиг. 9)

A screen shot of a computer screen

AI-generated content may be incorrect.

(фиг. 10)

## Задача 3:

A screen shot of a computer

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.

(фиг. 11-13)

Примерни резултати за for, while, iterator в тази последователност.

## Задача 4:

A computer screen shot of a program code

AI-generated content may be incorrect.

(фиг. 14)

Тук съм добавил и ASCII кода на space – „ “, за да не брои колко шпации има.

A black rectangular object with white lines

AI-generated content may be incorrect.

(фиг. 15)

Ако символът се повтаря, повече от веднъж се принтира.

## Задача 5:

A screen shot of a computer program

AI-generated content may be incorrect.

(фиг. 16)

A screenshot of a computer program

AI-generated content may be incorrect.

(фиг. 17)

Тези файлове се съхраняват в resources. При всяка грешка връщаме празен масив. Като записваме във файла сме означили header – average\_price.

A screenshot of a black background with numbers

AI-generated content may be incorrect.  
(фиг. 18-19)