Lab 11 – Functional Programming and Stream API

Objectives:

- Learn to create stream from a file or collection
- Learn to transform stream elements using Lambda function or Method Reference
- Learn to collect elements from stream

1. The file **MarkTwain.utf8.txt** contains a collection of works by Mark Twain, a famous American writer. You are asked to do text mining using Java Stream API. All tasks described below must be done with one single processing pipeline.

    a. Print out the top 10 most frequently used words in descending order.

```
run:
the=155357
and=122642
of=79507
a=73628
to=71719
it=51428
in=48204
i=48100
that=40846
was=39606
BUILD SUCCESSFUL (total time: 1 second)
```

    b. Group words by their initial in parallel and print out the frequent count of words from a to z.

```
run:
a=353873
b=128161
c=112568
d=88133
e=54506
f=101768
g=58234
h=184450
i=217871
j=14645
k=18011
l=70394
m=115386
n=71192
o=173020
p=81947
q=4680
r=55727
s=212220
t=453102
u=32733
v=16856
w=206413
x=682
y=41562
z=468
BUILD SUCCESSFUL (total time: 1 second)
```

    c. Find the top 10 most frequently used pair of words in descending order and save the result to a file.

```
1  of the=18241
2  in the=13406
3  it was=7537
4  to the=7121
5  and the=7031
6  it is=5929
7  he was=4553
8  on the=4467
9  of a=4367
10 was a=4271
```

Note that, when processing the words, they are treated **case-insensitively**. Also, all punctuations must be removed. For example, **after-dinner** is considered as two words **after** and **dinner**. Similarly, **you're** is treated as **you** and **re**.

- END -