

2021/12/10 上午3:09	WordPairCount.java	2021/12/10 上午3:09	WordPairCount.java
1 /*		53	* the concurrent access to prev.
2 *	To change this license header, choose License Headers in Project Properties.	54	*/
3 *	To change this template file, choose Tools Templates	55	String[] prev = { "" };
4 *	and open the template in the editor.	56	
5 /*		57	lines.flatMap(line -> Arrays.stream(line.replaceAll("[^a-zA-Z]", "
6 package lab11.solution;		58).toLowerCase().split("\\s+"))
7		59	.filter(w -> !w.isEmpty()).map(w -> {
8 import java.io.File;		60	Map.Entry<String, String> pair = Map.entry(prev[0], w);
9 import java.io.IOException;		61	prev[0] = w;
10 import java.io.PrintWriter;		62	return pair;
11 import java.net.URI;		63	}).collect(groupingBy(Function.identity(),
12 import java.net.URISyntaxException;		64	counting()))).entrySet().stream()
13 import java.nio.file.Files;		65	.sorted(Map.Entry.comparingByValue(Comparator.reverseOrder()))).limit(10).peek(System.
14 import java.nio.file.Path;		66	out::println)
15 import java.nio.file.Paths;		67	.forEach(
16 import java.util.Arrays;		68	e -> writer.printf("%s %s=%d\\n", e.getKey().getKey(),
17 import java.util.Comparator;		69	e.getValue());
18 import java.util.Map;		70	e.getKey().getValue(), e.getValue());
19 import java.util.Function;		71	
20 import java.util.logging.Level;		72	} catch (IOException ex) {
21 import java.util.logging.Logger;		73	Logger.getLogger(WordPairCount.class.getName()).log(Level.SEVERE, null,
22 import static java.util.stream.Collectors.counting;			ex);
23 import static java.util.stream.Collectors.groupingBy;			
24 import java.util.stream.Stream;			
25			
26 /**			
27 *	@author Wanting		
28 *			
29 /*			
30 public class WordPairCount {			
31			
32	public static void main(String[] args) throws URISyntaxException {		
33			
34	URI file = WordCount.class.getResource("given/MarkTwain.utf8.txt").toURI();		
35	Path path = Paths.get(file);		
36			
37	try (Stream<String> lines = Files.lines(path);		
38	PrintWriter writer = new PrintWriter(new File("paircount.txt"))) {		
39			
40	/*		
41	* Note that a lambda can only access free variables (the variables in		
42	its		
43	* enclosing context) that are final or effectively final.		
44	* The variable prev serves as a holder object and is effectively final.		
45	The		
46	final) but		
47	* the object is mutable.		
48	* Using such a "hack" will induce side effect to the program and is not		
49	a good		
50	all.		
51	* The attempt here does not cause problem because the stream is a		
52	sequential		
	* stream. For a parallel stream, the result will become unpredictable		
	due to		
localhost:4649/?mode=click	localhost:4649/?mode=click	localhost:4649/?mode=click	localhost:4649/?mode=click
1/2	2/2		