**Java 8 programs**

**Iteration of map:**

|  |
| --- |
| **// 1. Iterate map object using forEach**  Map<String, Integer> items = **new** HashMap<>();  items.put("A", 10);  items.put("B", 20);  items.put("C", 30);  items.put("D", 40);  items.put("E", 50);  items.put("F", 60);    //items.forEach((k,v) -> System.out.println(k +" : "+ v));  items.forEach((k,v) -> {  **if**(v > 40) {  System.***out***.println(k +" : "+ v);  }  });  **o/p:**  E : 50  F : 60  **// 2. Stream of skip and forEach**  IntStream.*range*(1, 10)  .skip(5)  .forEach(System.***out***::println);  **o/p:**  6  7  8  9  **// 3. Stream of sum of numbers**  **int** sum = IntStream.*range*(1, 10)  .sum();  System.***out***.println(sum);  **o/p:**  45  **// 4. Stream of sortd and findFirst**  Stream.*of*("ab", "nag", "scala", "java","mongodb")  .sorted()  .findFirst()  .ifPresent(System.***out***::println);  **// 5. Stream from array, filter,sorted and print**  String[] str = {"ab", "nag", "scala", "java","mongodb", "spring"};  Arrays.*stream*(str)  .filter( s -> s.startsWith("s"))  .sorted()  .forEach(System.***out***::println);  **// 6. Stream of map average and ifPresent**  Arrays.*stream*(**new** **int**[] {2,4,6,8,10})  .map((e) -> e \* e)  .average()  .ifPresent(System.***out***::println);    **int** sum = Arrays.*stream*(**new** **int**[] {2,4,6,8,10})  .map(e -> e\*e)  .sum();  System.***out***.println("sum: "+ sum);  **// 7. Stream of sorted,filter and foreach**  List<String> list = Arrays.*asList*("ab", "nag", "scala", "java","mongodb", "spring");  list.stream().map(String::toUpperCase)  .sorted()  .filter(e -> e.length() > 3)  .forEach(System.***out***::println);  **// 8. Stream rows from files, sorted, filter and print**  demo.txt  nagendra  scala good programing language  java best programing  rabbitmq  mongodb  swagger  spring  cloud  Stream<String> filedata = Files.*lines*(Paths.*get*("demo.txt"));  filedata.sorted()  .filter(e -> e.length() > 15)  .forEach(System.***out***::println);  filedata.close();  **o/p:**  java best programing  scala good programing language  **// 9. . Stream rows from files, filter and save into list**  Stream<String> filedata = Files.*lines*(Paths.*get*("demo.txt"));  List<String> list = filedata.sorted()  .filter( s -> s.contains("spring"))  .collect(Collectors.*toList*());  list.forEach(System.***out***::println);  filedata.close();  o/p  spring  // **10. Stream rows from files map,split,filter,count**  A,1,2,3  B,4,5,6  C,7,8,9  D,10,15  E  F,16,17,18,19,20  Stream<String> filedata = Files.*lines*(Paths.*get*("demo.txt"));  **long** count = filedata.map( e -> e.split(","))  .filter(e -> e.length == 4)  .count();  System.***out***.println(count);  o/p:  3  // **11. Stream rows from files map,split,filter,print**  Stream<String> filedata = Files.*lines*(Paths.*get*("demo.txt"));  filedata.map( e -> e.split(","))  .filter(e -> e.length == 4)  .filter(e -> Integer.*parseInt*(e[1]) > 6)  .forEach(x -> System.***out***.println(x[0] +" " + x[1] +" "+ x[2] + " "+ x[3]));  filedata.close();  o/p:  C 7 8 9  // **12. Stream rows from files map,split,filters, save into map**  Stream<String> filedata = Files.*lines*(Paths.*get*("demo.txt"));  Map<String, Integer> map = **new** HashMap<>();  map = filedata.map( e -> e.split(","))  .filter(e -> e.length == 4)  .filter(e -> Integer.*parseInt*(e[1]) > 2)  .collect(Collectors.*toMap*( k -> k[0], v -> Integer.*parseInt*(v[1])));  filedata.close();    **for**(Map.Entry<String, Integer> m : map.entrySet()) {  System.***out***.println(m.getKey() +" : "+ m.getValue());  }  o/p:  B : 4  C : 7  // **13. Stream of reduce**  **double** result = Stream.*of*(1.0,2.0,3.0,4.0,5.0).reduce(0.0, (Double a, Double b) -> a + b);  System.***out***.println(result);  o/p:  15.0  // **13. Stream of summarystatistics**  IntSummaryStatistics summary = IntStream.*of*(1,2,3,4,5)  .summaryStatistics();  System.***out***.println(summary);  o/p  IntSummaryStatistics{count=5, sum=15, min=1, average=3.000000, max=5} |