1. **Functional Interface**

An Interface that contains exactly one abstract method is known as functional interface. It can have any number of defaults, static methods but can contain only one abstract method.

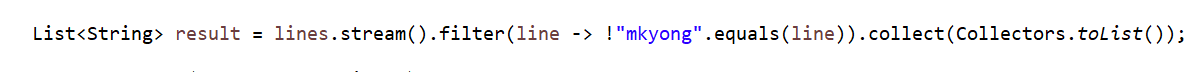
* A functional interface can extends another interface only when it does not have any abstract method.

1. **Lambda Expression**

* The Lambda expression is used to provide the implementation of an interface which has functional interface.
* Java lambda expression is treated as a function, so compiler does not create .class file.
* list.forEach((n)->System.out.println(n));

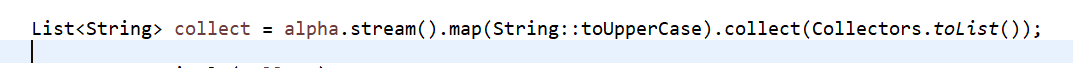
1. **Stream Filter**

**stream().filter()**



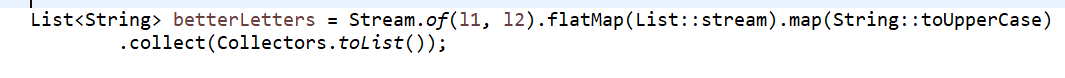
**stream().map()**

let’s you convert an object to something else.



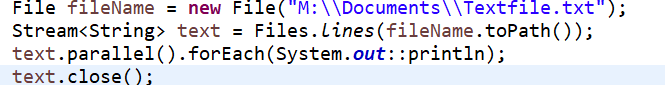
**flatMap()**

Is the combination of a map and a flat operation i.e, it first applies map function and then flattens the result

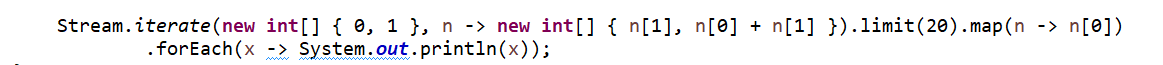


**parallel()**

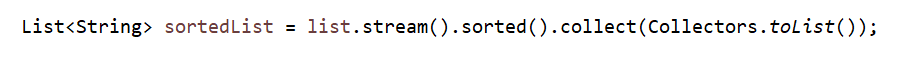
we can divide the code into multiple streams that are executed in parallel on separate cores and the result is the combination of the individual outcomes.



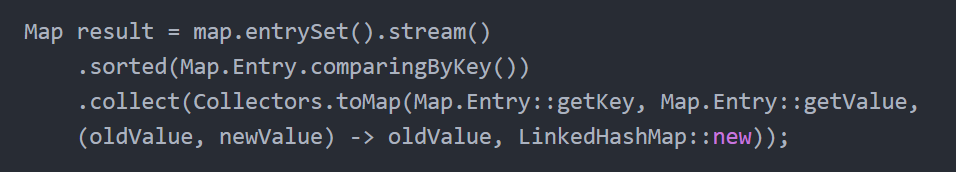
**Stream.iterate** to create stream values on demand, so called infinite stream.



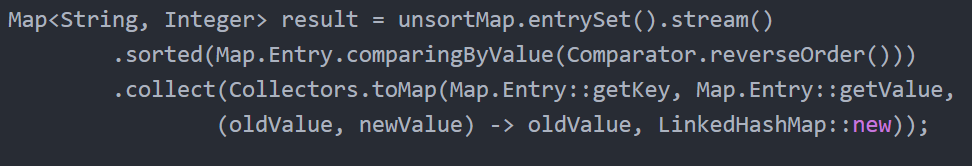
**Sorted()**



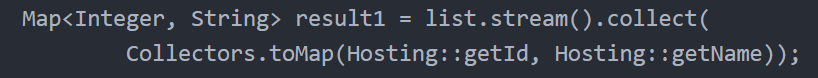
**Sort map by key**



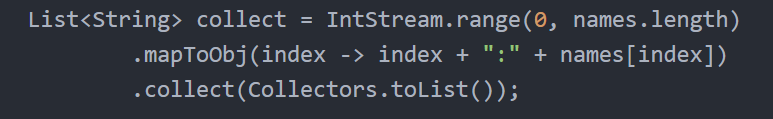
**Sort map by value**



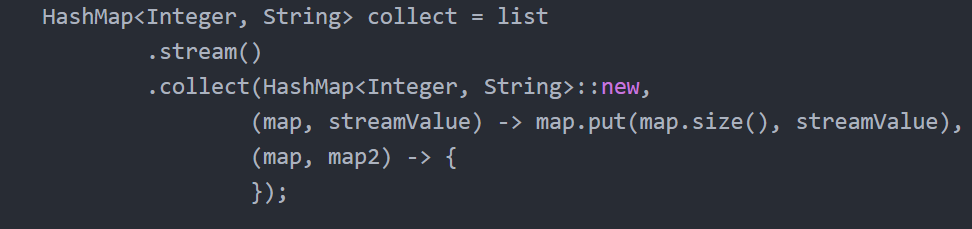
**List to map**



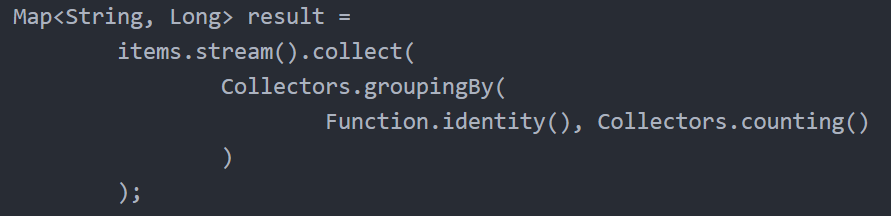
**foreach-print-with-index**



**List-with-index**



**Group by**



1. **S**
2. **S**
3. **S**
4. **D**
5. **D**
6. **d**