package com;

import java.util.Arrays;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

import java.util.function.Function;

import java.util.stream.Collectors;

public class CountAndSort {

public static void main(String[] args) {

// TODO Auto-generated method stub

List<String> items = Arrays.asList("Java","Python","Ruby","Java","Ruby","Python","Ruby");

Map<String,Long> result =items.stream().collect(Collectors.groupingBy(Function.identity(),Collectors.counting()));

Map<String,Long> finalMap = new LinkedHashMap<>();

result.entrySet().stream().sorted(Map.Entry.<String,Long>comparingByValue().reversed()).forEachOrdered(e->finalMap.put(e.getKey(),e.getValue()));

System.out.println(finalMap);

}

}

/\*\*\* OUTPUT

\* {Ruby=3, Java=2, Python=2}

\*/

/\*\*\*\*\*\*\*\*\*\*\* Theory of the functions used in above program \*\*\*\*\*\*/

/\* sorted() : Sorted returns a stream consisting of the elements ,according to natural order

\* comparingByValue(): Returns a omparator that compares Map.Entry in natural order on value

\* reversed() : To sort elements in descending order

\* forEachOrdered(): Method to traverse all the elements and perfrom action on each element of this stream

\* LinkedHashMap : Preserves ordering of elements in which they were inserted

\*/