Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 6 REPORT

STUDENT NAME Sıla Bengisu Yüksekli
STUDENT NUMBER 1801042877

Course Assistant: Doç. Dr. Erchan APTOULA

1 INTRODUCTION

1.1 Problem Definition

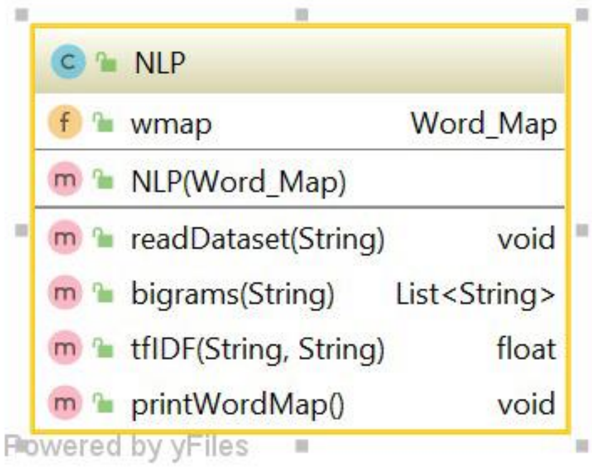
In this problem, some input file is given and we have to print an output according to this input file. This file consists of two queries which are revealing the bi-grams and calculating the TFIDF. Bi-gram is a text that contains a pair of words and this pair contains no words between them. They are sequential. So for example, if you have text like this “ This is beautiful” ,
Bi-grams will be like:
This is
İs beautiful.
TFIDF is a number that indicates the importamce of a given word. As a result in this problem, our aim are finding the bi-grams and calculating the TFIDF of a given file and printing the output.

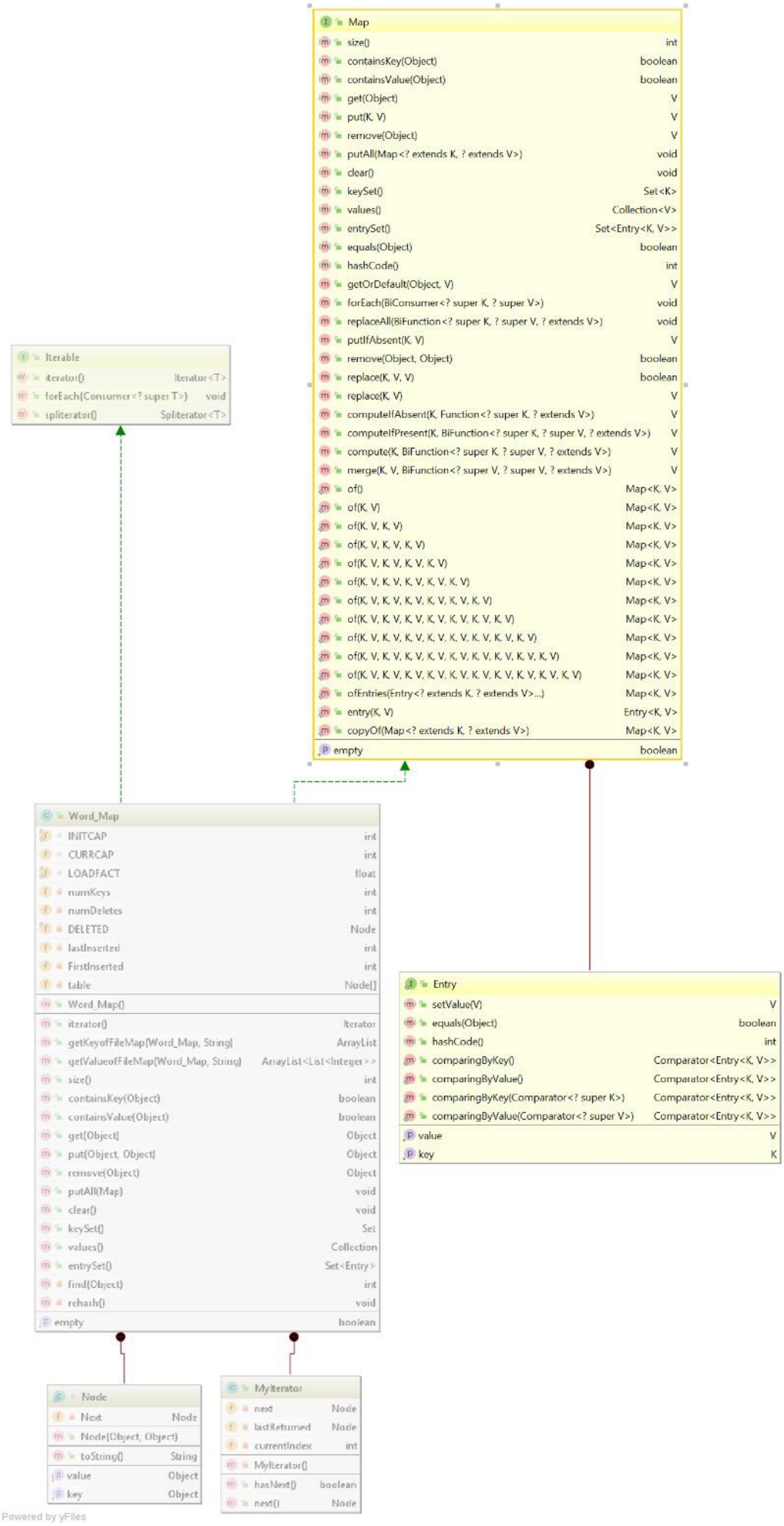
1.2 System Requirements

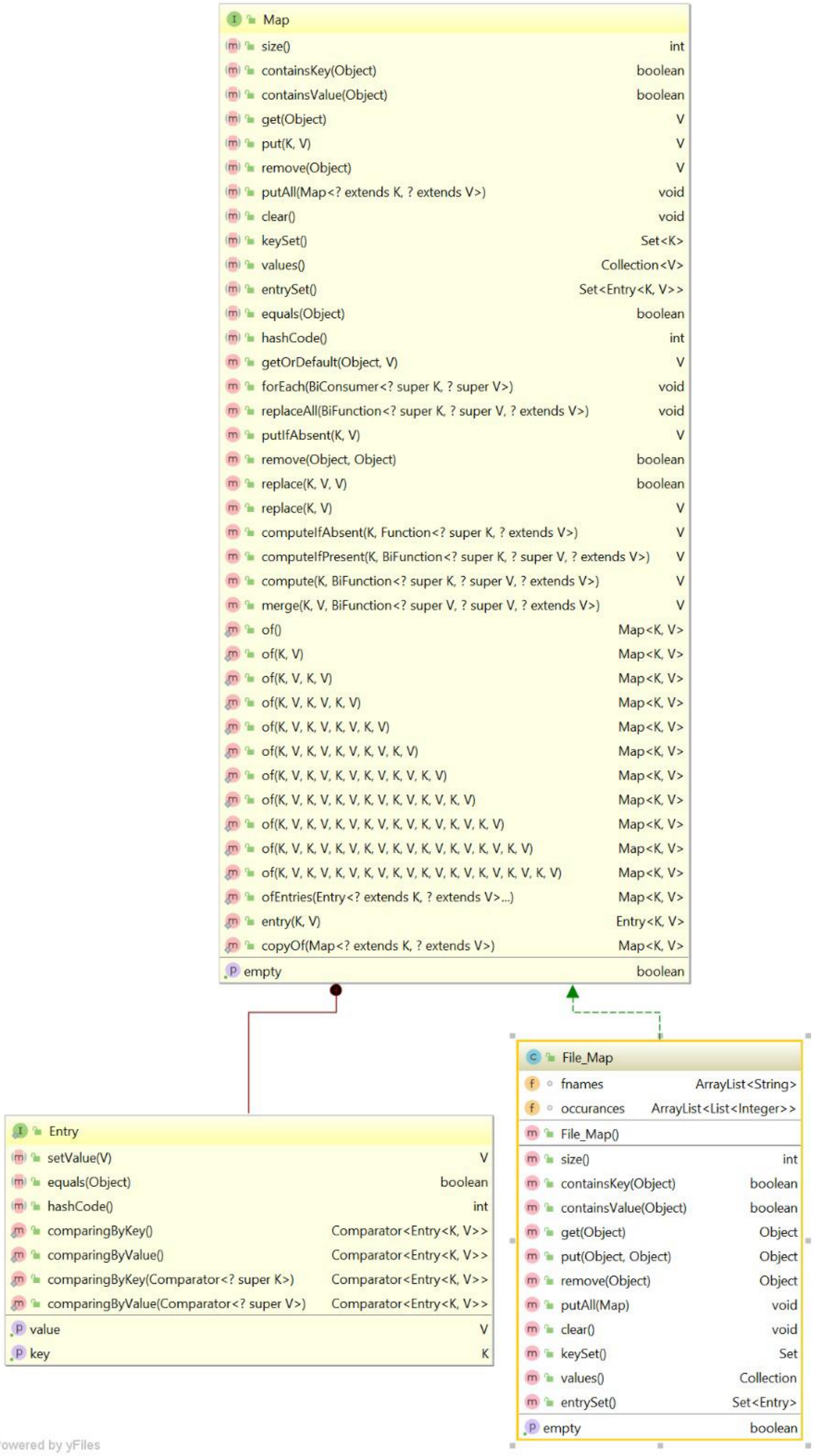
You can run this program everywhere which has java virtual machine. But if you want to use IntelliJ like me, maybe you need some requirements like minimum 1 GB RAM, 300 MB hard disk space, minimum 1 GB for caches, 1024x768 minimum screen resolutionSystem requirements.

2 METHOD

2.1 Class Diagrams




Map
size() int
containsKey(Object) boolean
containsValue(Object) boolean
get(Object) V
put(K, V) V
remove(Object) V
putAll(Map<? extends K, ? extends V>) void
clear() void
keySet() Set<K>
values() Collection<V>
entrySet() Set<Entry<K, V>>
equals(Object) boolean
hashCode() int
getOrDefault(Object, V) V
forEach(BiConsumer<? super K, ? super V>) void
replaceAll(BiFunction<? super K, ? super V, ? extends V>) void
putIfAbsent(K, V) V
remove(Object, Object) boolean
replace(K, V, V) boolean
replace(K, V) V
computeIfAbsent(K, Function<? super K, ? extends V>) V
computeIfPresent(K, BiFunction<? super K, ? super V, ? extends V>) V
compute(K, BiFunction<? super K, ? super V, ? extends V>) V
merge(K, V, BiFunction<? super V, ? super V, ? extends V>) V
of() Map<K, V>
of(K, V) Map<K, V>
of(K, V, K, V) Map<K, V>
of(K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V) Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V) Map<K, V>
ofEntries(Entry<? extends K, ? extends V>...) Map<K, V>
entry(K, V) Entry<K, V>
copyOf(Map<? extends K, ? extends V>) Map<K, V>
empty boolean
Iterable
iterator() Iterator<T>
forEach(Consumer<? super T>) void
spliterator() Spliterator<T>
Word_Map
INITCAP int
CURRCAP int
LOADFACT float
numKeys int
numDeletes int
DELETED Node
lastInserted int
FirstInserted int
table Node[]
Word_Map()
iterator() Iterator
getKeyofFileMap(Word_Map, String) ArrayList
getValueofFileMap(Word_Map, String) ArrayList<List<Integer>>
size() int
containsKey(Object) boolean
containsValue(Object) boolean
get(Object) Object
put(Object, Object) Object
remove(Object) Object
putAll(Map) void
clear() void
keySet() Set
values() Collection
entrySet() Set<Entry>
find(Object) int
rehash() void
empty boolean
Entry
setValue(V) V
equals(Object) boolean
hashCode() int
comparingByKey() Comparator<Entry<K, V>>
comparingByValue() Comparator<Entry<K, V>>
comparingByKey(Comparator<? super K>) Comparator<Entry<K, V>>
comparingByValue(Comparator<? super V>) Comparator<Entry<K, V>>
value V
key K
Node
Next Node
Node(Object, Object)
toString() String
value Object
key Object
MyIterator
next Node
lastReturned Node
currentIndex int
MyIterator()
hasNext() boolean
next() Node
Powered by yFiles



Map	
size()	int
containsKey(Object)	boolean
containsValue(Object)	boolean
get(Object)	V
put(K, V)	V
remove(Object)	V
putAll(Map<? extends K, ? extends V>)	void
clear()	void
keySet()	Set<K>
values()	Collection<V>
entrySet()	Set<Entry<K, V>>
equals(Object)	boolean
hashCode()	int
getOrDefault(Object, V)	V
forEach(BiConsumer<? super K, ? super V>)	void
replaceAll(BiFunction<? super K, ? super V, ? extends V>)	void
putIfAbsent(K, V)	V
remove(Object, Object)	boolean
replace(K, V, V)	boolean
replace(K, V)	V
computeIfAbsent(K, Function<? super K, ? extends V>)	V
computeIfPresent(K, BiFunction<? super K, ? super V, ? extends V>)	V
compute(K, BiFunction<? super K, ? super V, ? extends V>)	V
merge(K, V, BiFunction<? super V, ? super V, ? extends V>)	V
of()	Map<K, V>
of(K, V)	Map<K, V>
of(K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V)	Map<K, V>
of(K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V, K, V)	Map<K, V>
ofEntries(Entry<? extends K, ? extends V>...)	Map<K, V>
entry(K, V)	Entry<K, V>
copyOf(Map<? extends K, ? extends V>)	Map<K, V>
empty	boolean

Entry	
setValue(V)	V
equals(Object)	boolean
hashCode()	int
comparingByKey()	Comparator<Entry<K, V>>
comparingByValue()	Comparator<Entry<K, V>>
comparingByKey(Comparator<? super K>)	Comparator<Entry<K, V>>
comparingByValue(Comparator<? super V>)	Comparator<Entry<K, V>>
value	V
key	K

File_Map	
fnames	ArrayList<String>
occurances	ArrayList<List<Integer>>
File_Map()	
size()	int
containsKey(Object)	boolean
containsValue(Object)	boolean
get(Object)	Object
put(Object, Object)	Object
remove(Object)	Object
putAll(Map)	void
clear()	void
keySet()	Set
values()	Collection
entrySet()	Set<Entry>
empty	boolean

2.2 Use Case Diagrams

The user can open the file in Intellij. First click the File button which is located at the left-up edge and chose open. After some window opened, choose the project. Then setup the SDK. Run the project from the green button which is located at right-up edge, but if user want to run in some other place, if there is java virtual machine, he/she can do.

2.3 Problem Solution Approach

In this problem some input file is given and I had to read the words from this file. Every words has a some kind of filemap and this map indicates the files where the words occurs and the positions in that file. After that I had to calculate the TFIDF and find the bi-grams. So the best way to do this is using map structure. Because I must store the words and every word can be in a another place. For the filemap I used ArrayList to store keys and values. For filemap Arraylist is a suitable choice, because words can be different positions in the same file. So I might have extend the list because of this situation and when you use ArrayList, you don't have worry about reallocation.

Time complexity for the FileMap methods

-***size:*** O(1) because I use the size method and the complexity of size method is O(1).
-***isEmpty:*** O(1), because I use the isEmpty method of arrayList and the complexity of this O(1).
-containsKey: *O(n) , because indexOf method is O(n).*

-containsValue: O(n), because there is while loop im this method.

-get: O(n), because indexOf method is O(n)

-put: O(n), because indexOf method is O(n), add and get methods is O(1).

-remove: O(n), because indexOf method is O(n) and remove is O(n).

-clear: O(n), because of the clear method in the ArrayList

-keySet:O(n), because there is a while loop.

-values: O(n) because of the addAll method.

Time complexity for the WordMap methods

-***size:*** O(1), It just returns the number of the keys.
-***isEmpty:*** O(1), It just returns true if the number of the keys equals to zero.
-containsKey: *O(n), because there is one while loop in it.*

-containsValue: O(n), because there is one while loop in it.

-get: O(n), because it enters find method which has a one while loop

-put: O(n), because it enters find method and it has one while loop.

-clear: O(n), because it has one while loop.

-keySet: O(n), because it has one while loop.

-values: O(n), because it has one while loop.

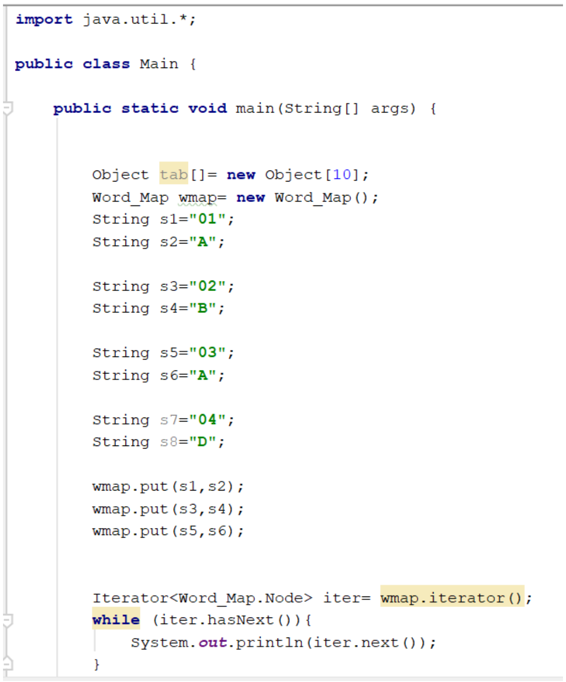
-rehash: O(n), because it has one for loop.

3 RESULT

3.1 Test Cases

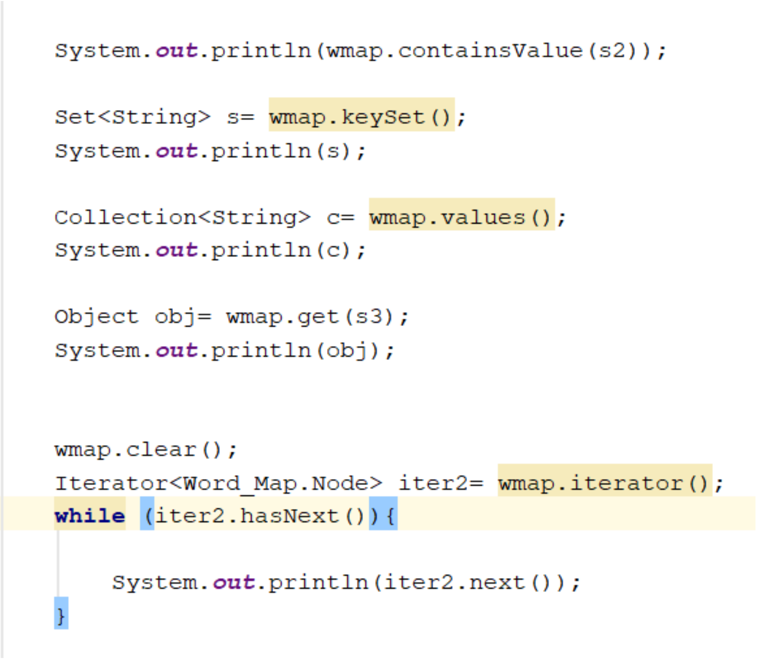
First, I tested WordMap class in main repeatedly by using some arbitary String values and Keys as you can see in the screenshots. I tested all the method of this class by checking the resut of compilation. Then, I do the same things for FileMap class. I choose arbitary String values and List keys and checked all the functionality of methods of this class.

3.2 Running Results



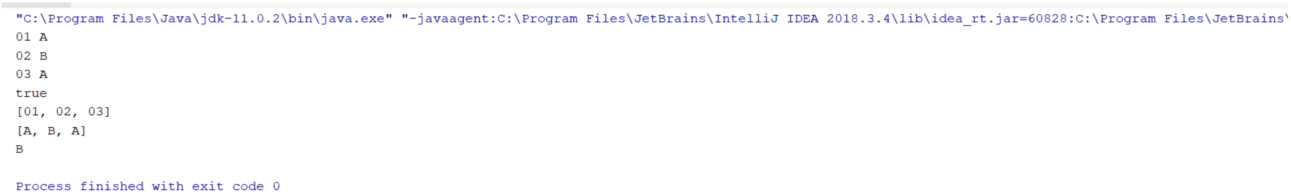
```
import java.util.*;

public class Main {

    public static void main(String[] args) {


        Object tab[]= new Object[10];
        Word_Map wmap= new Word_Map();
        String s1="01";
        String s2="A";

        String s3="02";
        String s4="B";

        String s5="03";
        String s6="A";

        String s7="04";
        String s8="D";

        wmap.put(s1,s2);
        wmap.put(s3,s4);
        wmap.put(s5,s6);


        Iterator<Word_Map.Node> iter= wmap.iterator();
        while (iter.hasNext()){
            System.out.println(iter.next());
        }
```

Test for methods of Word_Map(for general case)



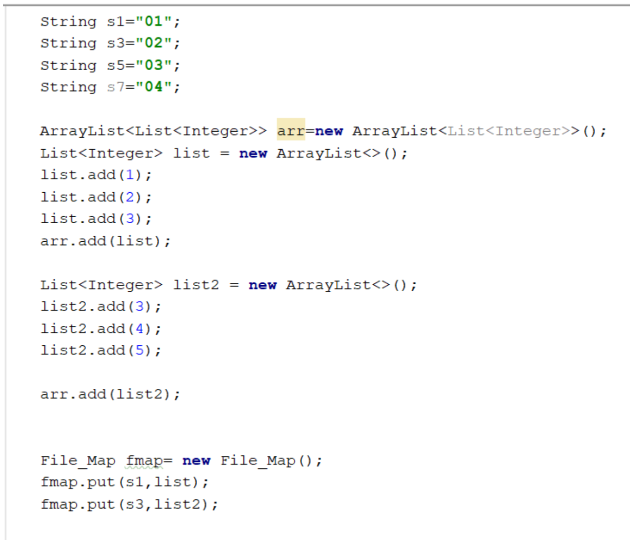
```
System.out.println(wmap.containsValue(s2));

Set<String> s= wmap.keySet();
System.out.println(s);

Collection<String> c= wmap.values();
System.out.println(c);

Object obj= wmap.get(s3);
System.out.println(obj);


wmap.clear();
Iterator<Word_Map.Node> iter2= wmap.iterator();
while (iter2.hasNext()){

    System.out.println(iter2.next());
}
```

Testing for methods of Word_Map – part2



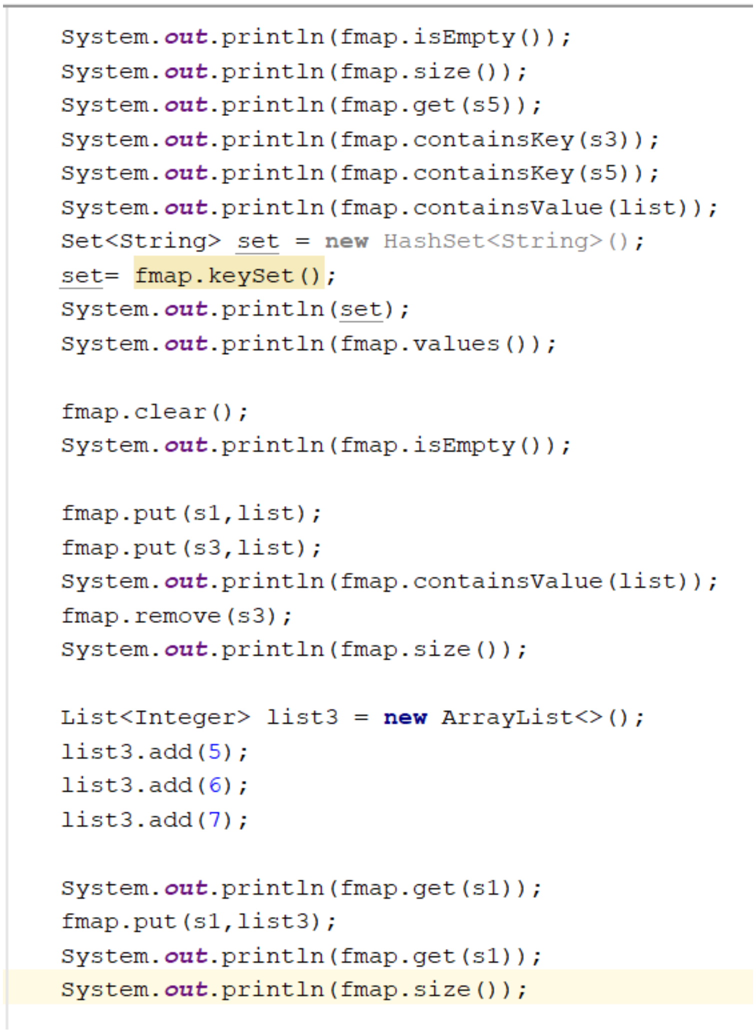
```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2018.3.4\lib\idea_rt.jar=60828:C:\Program Files\JetBrains\
01 A
02 B
03 A
true
[01, 02, 03]
[A, B, A]
B

Process finished with exit code 0
```

Result



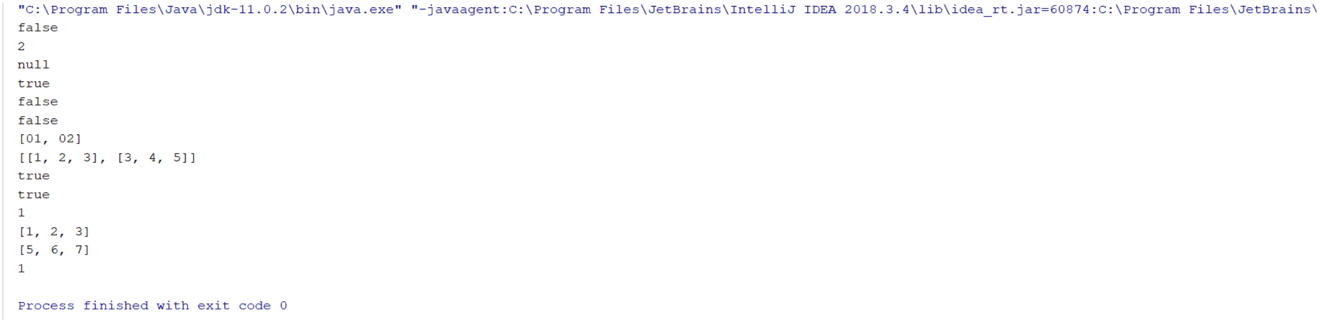
```
String s1="01";
String s3="02";
String s5="03";
String s7="04";

ArrayList<List<Integer>> arr=new ArrayList<List<Integer>>();
List<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.add(3);
arr.add(list);

List<Integer> list2 = new ArrayList<>();
list2.add(3);
list2.add(4);
list2.add(5);

arr.add(list2);


File_Map fmap= new File_Map();
fmap.put(s1,list);
fmap.put(s3,list2);
```

Testing For methods of File_Map (general case)



```
    System.out.println(fmap.isEmpty());
    System.out.println(fmap.size());
    System.out.println(fmap.get(s5));
    System.out.println(fmap.containsKey(s3));
    System.out.println(fmap.containsKey(s5));
    System.out.println(fmap.containsValue(list));
    Set<String> set = new HashSet<String>();
    set= fmap.keySet();
    System.out.println(set);
    System.out.println(fmap.values());

    fmap.clear();
    System.out.println(fmap.isEmpty());

    fmap.put(s1,list);
    fmap.put(s3,list);
    System.out.println(fmap.containsValue(list));
    fmap.remove(s3);
    System.out.println(fmap.size());

    List<Integer> list3 = new ArrayList<>();
    list3.add(5);
    list3.add(6);
    list3.add(7);

    System.out.println(fmap.get(s1));
    fmap.put(s1,list3);
    System.out.println(fmap.get(s1));
    System.out.println(fmap.size());
```

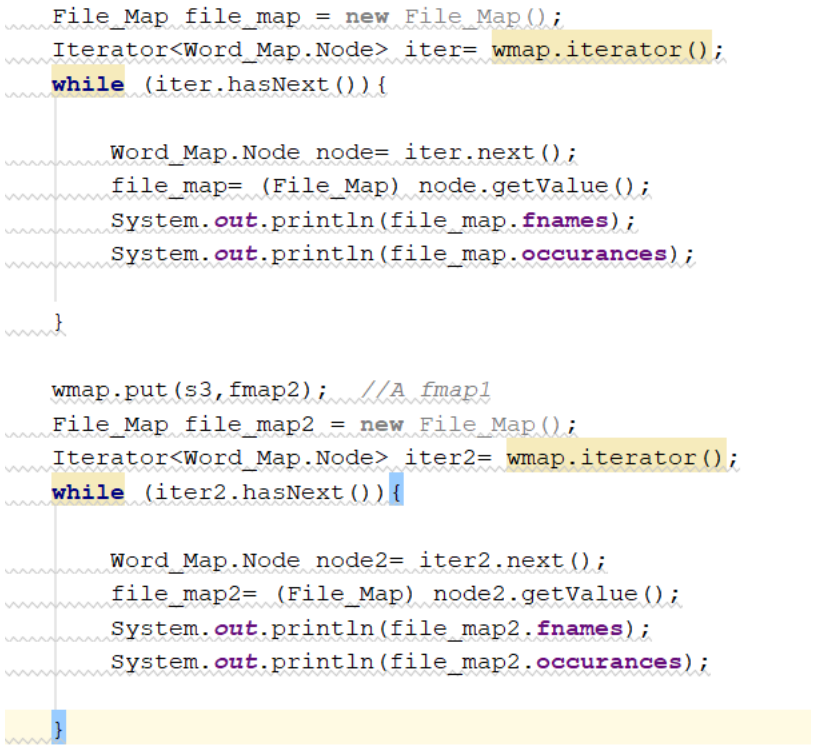
Testing For methods of File_Map – Part2



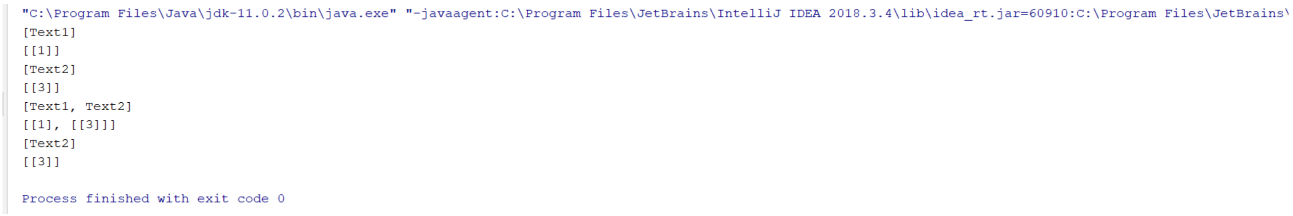
```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2018.3.4\lib\idea_rt.jar=60874:C:\Program Files\JetBrains\
false
2
null
true
false
false
[01, 02]
[[1, 2, 3], [3, 4, 5]]
true
true
1
[1, 2, 3]
[5, 6, 7]
1

Process finished with exit code 0
```

Result

Testing for our case



```
File_Map file_map = new File_Map();
Iterator<Word_Map.Node> iter= wmap.iterator();
while (iter.hasNext()){

    Word_Map.Node node= iter.next();
    file_map= (File_Map) node.getValue();
    System.out.println(file_map.fnames);
    System.out.println(file_map.occurances);

}

wmap.put(s3,fmap2);  //A fmap1
File_Map file_map2 = new File_Map();
Iterator<Word_Map.Node> iter2= wmap.iterator();
while (iter2.hasNext()){

    Word_Map.Node node2= iter2.next();
    file_map2= (File_Map) node2.getValue();
    System.out.println(file_map2.fnames);
    System.out.println(file_map2.occurances);

}
```



```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2018.3.4\lib\idea_rt.jar=60910:C:\Program Files\JetBrains\
[Text1]
[[1]]
[Text2]
[[3]]
[Text1, Text2]
[[1], [[3]]]
[Text2]
[[3]]

Process finished with exit code 0
```

Result