

Assignment 5

Do all same all program for linkedlist, hashset, and for hashmap which you did in arraylist. You have to make a 3 class file to execute all the method with respective linkedlist, hashset, and hashmap. Problem statement will be same but method will be used as per the linkedlist, hashset, and hashmap.

```

import java.util.*;
class hashmap
{
    public static void main (String args[])
    {
        HashMap<String, Integer> map = new HashMap<String, Integer>();
        map.put ("java", 1);
        map.put ("c", 20);
        map.put ("c++", 30);
        map.put ("ada", 40);
        Hashmap<String, Integer> map2 = new Hashmap<String, Integer>();
        map2.put ("#", 50);
        map2.put ("python", 60);
        System.out.println ("Is the key '5' present?");
        map.containskey (5));
        System.out.println ("Is the key '5' present?");
        map.containskey (10));
    }
}

```

```
System.out.println("Is the value 'world' present?  
+map.containsKey("c"));  
System.out.println("Is the map empty? "+map.  
isEmpty());  
System.out.println("The set is: "+map.get());  
System.out.println("The value is: "+map.get());  
System.out.println("The set is: "+map.keySet());  
System.out.println("The size of the map is "+  
map.size()); map.put("JSE", 15);  
HashMap<String, Integer> new_hash_map  
= new HashMap<String, Integer>();  
new_hash_map.putAll(map);  
System.out.println("The cloned map look  
like this: "+map.clone());  
}
```

```
import java.util.*;  
class hashset  
{  
public static void main (String args[])  
{  
HashSet<String> arr = new HashSet<String>();  
arr.add ("collection framework in java");  
arr.add ("ArrayList");  
arr.add ("Vector");  
arr.add ("List");
```

```
HashSet<String> arr2 = new HashSet<String>();
arr2.add ("Linked List");
arr2.add ("vector");
System.out.println ("in Does set 1 contains set 2:");
+ arr.containsAll (arr2));
System.out.println ("Elements in hashset :" + arr);
System.out.println ();
boolean value = arr.equals (arr2);
// print the value
System.out.println ("Are both set equal :" + value);
boolean flag = arr.contains ("list");
if (flag == true){
    System.out.println ("hashset doesn't contain
element list");
    System.out.println ();
} else {
    System.out.println ("hashset doesn't contain
element list");
    System.out.println ();
}
arr.remove (2);
System.out.println ("hashset After removing
element at index 2 = " + arr);
System.out.println ();
HashSet cloned_set = new HashSet ();
cloned_set = (HashSet) arr.clone ();
```

```
1 System.out.println ("The new set :" + cloned.set);
System.out.println ("size of hashset is :" +
arr.size());
System.out.println ();
Iterator value2 = arr.iterator();
System.out.println ("the iterator values are:");
while (value2.hasNext ()) {
System.out.println ("value at index " + it
of Array converted from hashset = " + object [i]);
System.out.println ();
}
System.out.println ("HashCode value :" + arr.
hashCode ());
boolean flag1 = arr.isEmpty ();
if (flag1 == true) {
System.out.println ("ArrayList is Empty");
}
else {
System.out.println ("ArrayList is not Empty");
}
System.out.println ();
arr.clear ();
System.out.println ("empty arrayList after using
clear method :" + arr);
}
```

```
import java.util.*;  
1. class linkedlist  
{  
public static void main (String args [])  
{  
linkedlist<String> l1 = new linkedlist<String>();  
l1.add ("A");  
l1.add ("B");  
l1.addlast ("C");  
l1.addfirst ("D");  
l1.add (2, "F");  
System.out.println (l1);  
l1.remove (3);  
l1.removefirst ();  
l1.set (1, "F");  
for (String str : l1)  
System.out.println (str + " ");  
linkedlist sec.list = new linkedlist ();  
sec.list = (linkedlist) l1.clone ();  
System.out.println ("second linkedlist is:" +  
sec.list);  
System.out.println ("The object that is  
replaced is :" + l1.set (2, "G"));  
System.out.println ("The element is :" + l1.get  
(2));  
System.out.println ("Does the list contain  
'A' :" + l1.contains ("A"));
```

```
1. l1.push("z");
System.out.println(l1);
String s = l1.pop();
System.out.println(s);
Collection<String> collect = new ArrayList<String>();
collect.add("A");
collect.add("computer");
l1.addAll(collect);
System.out.println("The first element is:" + l1.getFirst());
System.out.println("The last element is:" + l1.getLast());
System.out.println("The first element is:" + l1.removeFirst());
System.out.println("The last element is:" + l1.removeLast());
l1.addLast("Last");
l1.add("L");
l1.add("M");
l1.add("N");
System.out.println("The list is as follows:");
ListIterator listIter = l1.listIterator(2);
while (listIter.hasNext()){
    System.out.println(listIter.next());
}
System.out.println("The first occurrence of")
```

1)

F is at index: "+ l1.indexOf("F"));
l1.clear();

System.out.println("List after clearing all
elements: "+ l1);

}

Assignment 6

file I/O problem

Create new file 'test.txt'.

Write a data in it of about 300 words.

read a file 'test.txt'.

Calculate the letter in it. (a, b, c, d, ...)

Constant in that file (b, c, d, f, ...,)

Vowel in that file (a, e, i, o, u, ...)

words in a complete file ("JAVA")

Calculate how many time one character is

repeated (a=10, b=50, z=34, ...)

Save that file with other name 'test.out'.
txt'.

```
→ import java.io.*;
import java.util.*;
class createfile
{
    static final int MAX_CHAR = 256;
    static int countchar (String str)
    {
        int count5 [] = new int [MAX_CHAR];
        int len = str.length ();
        for (int i2 = 0; i2 < len; i2++)
            count5 [str.charAt (i2)]++;
        char ch2 [] = new char [str.length ()];
        for (int i2 = 0; i2 < len; i2++)
        {
```

```

chz[i2] = str.charAt(i2);
int find = 0;
for(int j2=0; j2 <= i2; j2++) {
    if (str.charAt(j2) == chz[j2]) find++;
}
if (find == 1)
    System.out.println("Number of occurrence of"
    str.charAt(i2) + " is :" + count5[str.charAt(i2)]);
}
return 0;
}

```

Public static void main (String [] args) throws
 IOException {
 // Accept a String
 String str = "File Handling in Java using "+
 "FileWriter and FileReader in Java "+"jav
 FileWriter and FileReader classes are use
 to write and read data from text file
 they are character stream classes it is recom
 ended not to use the FileInputStream and
 OutputStream classes if you have to read
 write any textual information as these are
 Byte Stream classes "+" FileWriter is use
 to create a file writing characters into
 this class inherits from the OutputStream
 class The constructors of this class

assume that the default character encoding and the default byte buffer size are acceptable. To specify these values yourself construct an `OutputStreamWriter` on a `FileOutputStream` "+"

`FileWriter` is meant for writing streams of characters for writing streams of raw bytes consider using a `FileOutputStream` "+" constructors `FileWriter` file `fw` constructs "+" a `FileWriter` object given a file object `FileWriter` file `boolean append` the data writing "+" This class inherit from the class the constructors "+" of the class assume that the default character encoding and the default byte buffer size are appropriate to specify these values yourself construct an `InputStreamReader` on a `FileInputStream` stream "+" `FileReader` is meant for reading streams of characters "+" for reading streams of raw bytes consider using a `FileInputStream`

```
fileWriter fw = new FileWriter ("test.txt")
for (int i=0; i<str.length(); i++)
    fw.write (str.charAt(i))
System.out.println ("writing successful")
fw.close()
```

```
int ch;
```

```
fileReader fr = null;
```

```
try {
```

```
fr = new FileReader ("test.txt");
```

```

2) } catch (FileNotFoundException fe)
{ system.out.println ("file not found");
}

int vowels = 0, consonants = 0, p = 0;
while ((ch = fr.read ()) != -1)
{ // system.out.println ((char)ch);
}

int count = 0, count1 = 0;
char ch1;

str = str.toLowerCase ();
for (int k = 0; k < str.length (); k++)
{
    if (str.charAt (k) == '.')
    {
        count++;
    }

    if (str.charAt (k) == 'a' || str.charAt (k) == 'e')
    str.charAt (k) == 'i' || str.charAt (k) == 'o'
    str.charAt (k) == 'u') {
        vowels++;
    }

    else {
        consonants++;
    }
}

char ch2[] = new char [str.length ()];
for (int l = 0; l < str.length (); l++)
{
}

```

3)
 ch2[i] = str.charAt(i);
 if ((i > 0) && (ch2[i] != '.') && (ch2[i - 1] == '.'))
 || ((ch2[0] != '.') && (i == 0)))
 count++;
 System.out.println("Total number of characters
 in a string:" + count);
 System.out.println("Total number of vowels in
 test.txt file is:" + vowels);
 System.out.println("Total number of constant
 in test.txt file is:" + constant);
 System.out.println("Total number of words in
 test.txt file is:" + constants);
 System.out.println("Total number of words
 in test.txt file is:" + count);
 foundchar(str);
 fr.close();
 }
 }

Assignment 7

Generic

1. 1. write a generic method to find the max.
- i/p : {2, 62, 4, 78, 6, 10, 49, 20, 59, 43, 29, 30, 56, 89}
- out : 89

```

import java.io.*;
import java.util.*;

class newex
{
    public static <T extends Object & Comparable<? super T>> T getMax(List<? extends T>
list, int begin, int end)
{
    T Maxelem = list.get(begin);
    for (++begin; begin < end; ++begin)
        if (Maxelem.compareTo(list.get(begin)) < 0)
            Maxelem = list.get(begin);
    return Maxelem;
}

public static void main(String args[])
{
    List<Integer> arr = Arrays.asList(2, 62, 4, 78,
                                     6, 10, 49, 20, 59, 43, 29, 30, 56, 89);
    int x = newex.getMax(arr, 0, arr.size());
    System.out.println("maximal number :" + x);
}

```

2. write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, even numbers)

i/p: {2, 4, 6, 7, 8, 9, 90, 78, 41, 56, 79, 45, 65, 85}

Output :

even : 7

odd : 7

```
import java.io.*;
import java.util.*;
class Algorithm {
    public static <T> int countIf(Collection<T> c, UnaryPredicate<T> p) {
        int count = 0;
        for (T elem : c)
            if (p.test(elem))
                ++count;
        return count;
    }
}
```

```
interface UnaryPredicate<T> {
    public boolean test(T obj);
}
```

```
class OddPredicate implements UnaryPredicate<Integer> {
```

public boolean test (Integer i) { return i % 2 == 0; } }

class EvenPredicate implements UnaryPredicate<Integer> {
 public boolean test (Integer i) { return i % 2 == 0; } }

public class NewEx2 {
 public static void main (String [] args) {
 Collection<Integer> ci = Arrays.asList (1, 2, 3, 4);
 int count = Algorithm.countIf (ci, new OddPredicate());
 int count2 = Algorithm.countIf (ci, new EvenPredicate());
 System.out.println ("Number of odd integers=" + count);
 System.out.println ("Number of even integers=" + count2);
 } }