**Lists**:

The List is probably the single most useful and widely used type of Collection. List is a general interface, and ArrayList and LinkedList are implementing classes. ArrayList is the best general-purpose List. A List is a linear structure where each element is known by an index number 0, 1, 2, ... len-1 (like an array). Lists can only store objects, like String and Integer, but not primitives like int. You can’t create a List of int, but you can create a list of Integer objects. This is a common feature of all the Java Collection classes. Another way to say this is that the collection classes can only store pointers.

**Implements:**

* ***Reference* *Types***
* ***Numbers***
* ***Duplicates***
* ***Items in order.***

**import java.util.ArrayList;**

**import java.util.HashSet;**

**import java.util.List;**

**import java.util.Set;**

**public class ConvertListToSet**

**{**

**public static void main( String[] args )**

**{**

**System.out.println("List values .....");**

**List<String> list = new ArrayList<String>();**

**list.add("1");**

**list.add("2");**

**list.add("3");**

**list.add("4");**

**list.add("1");**

**for (String temp : list){**

**System.out.println(temp);**

**}**

**Set<String> set = new HashSet<String>(list);**

**System.out.println("Set values .....");**

**for (String temp : set){**

**System.out.println(temp);**

**}**

**}**

**}**

**Sets:**

The Java "Set" is a Collection, like the List, but with the added constraint that each element can be in

the Set once, using equals() to test if two elements are the same. Sets are potentially more efficient than lists. In particular, the HashSet can find or insert an element in constant time. Set is an interface that extends Collection. It is an unordered collection of objects in which duplicate values cannot be stored.

**Implements:**

* ***Add***
* ***Remove***
* ***Clear***

import java.util.\*;

public class Set\_example

{

    public static void main(String[] args)

    {

        Set<String> hash\_Set = new HashSet<String>();

        hash\_Set.add("Geeks");

        hash\_Set.add("For");

        hash\_Set.add("Geeks");

        hash\_Set.add("Example");

        hash\_Set.add("Set");

        System.out.print("Set output without the duplicates");

        System.out.println(hash\_Set);

        System.out.print("Sorted Set after passing into TreeSet");

        Set<String> tree\_Set = new TreeSet<String>(hash\_Set);

        System.out.println(tree\_Set);

    }

}

**Maps:**

A Map is an object that maps keys to values. A Map cannot contain duplicate keys. A Map is a key/value table that can look up any entry by key very efficiently (known as a "hash table" or "dictionary"). "Map" is a general interface of the basic map features, implemented by two main classes: HashMap and TreeMap. Both the key and value must be object types such as String or Integer or List. A Map stores key/value entries, where each key in the map is associated with a single value.

**Implements:**

* ***HashMap***
* ***TreeMap***
* ***LinkedHashMap***
* **Internally sorted by the index**

package CodingHmwrk;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

public class test {

public static void main(String[] args) {

Map<String, String> mapCountryCodes = new HashMap<>();

mapCountryCodes.put("1", "USA");

mapCountryCodes.put("44", "United Kingdom");

mapCountryCodes.put("33", "France");

mapCountryCodes.put("81", "Japan");

Set<String> setCodes = mapCountryCodes.keySet();

Iterator<String> iterator = setCodes.iterator();

while (iterator.hasNext()) {

String code = iterator.next();

String country = mapCountryCodes.get(code);

System.out.println(code + " => " + country);

}

}

}

**Codes:**

**//ArrayList of type String**

import java.util.ArrayList;

public class Program {

public static void main(String[] args) {

ArrayList<Integer> elements = new ArrayList<>();

elements.add(10);

elements.add(15);

elements.add(20);

int count = elements.size();

System.out.printIn(“Count: “ + count)

For (int i = 0; i < elements.size(); i++) {

int value = elements.get(i);

System.out.printIn(“Element: ” + value);

}

}

}

**//HashSet of type StringBuilder**

package CodingHmwrk;

import java.util.HashSet;

public class test {

public static void main(String[] args) {

}

public int countUniqueWords(String line) {

HashSet<String> hashset = new HashSet<String>();

StringBuilder word = new StringBuilder();

int endOfLine = line.length() - 1;

boolean isWord = false;

String stringWord = null;

Integer counter = 0;

for (int i = 0; i < line.length(); i++) {

if (Character.isLetter(line.charAt(i)) == true && i != endOfLine) {

word.append(line.charAt(i));

} else if (Character.isLetter(line.charAt(i)) == false && isWord == true) {

counter++;

stringWord = word.toString();

hashset.add(stringWord);

word = null;

isWord = false;

} else if (Character.isLetter(line.charAt(i)) && i == endOfLine) {

counter++;

stringWord = word.toString();

hashset.add(stringWord);

}

}

System.out.println(counter);

System.out.println(hashset.size());

System.out.println(hashset);

return counter;

}

}

**//HashMap of type String, String**

package CodingHmwrk;

import java.util.HashMap;

import java.util.Map.Entry;

public class test {

public static void main(String[] args) {

HashMap<String, String> map = new HashMap<String, String>();

map.put("key1", "value1");

map.put("key2", "value2");

map.put("key3", "value3")

System.out.println("using entrySet and toString");

for (Entry<String, String> entry : map.entrySet()) {

System.out.println(entry);

}

System.out.println();

System.out.println("using entrySet and manual string creation");

for (Entry<String, String> entry : map.entrySet()) {

System.out.println(entry.getKey() + "=" + entry.getValue());

}

System.out.println();

System.out.println("using keySet");

for (String key : map.keySet()) {

System.out.println(key + "=" + map.get(key));

}

System.out.println();

}

}

**References**:

<https://web.stanford.edu/class/archive/cs/cs108/cs108.1092/handouts/02SCollections.pdf>

<https://www.geeksforgeeks.org/set-in-java/>

<https://stackoverflow.com/questions/13117085/building-strings-for-a-hashset-in-java-unable-to-loop-correctly-through-large>

<https://stackoverflow.com/questions/35642493/mapstring-string-how-to-print-both-the-key-string-and-value-string-toget>

<http://www.codejava.net/java-core/collections/java-map-collection-tutorial-and-examples>