Plan for the day:

1. Java hands on practice for M1
2. Cheat sheet for Java
3. Remove if method in collection
4. Online coding challenges
5. String & Regex
6. Module 1 curriculum
7. Some of our participants are going to present their topics
8. Iterator to overcome ConcurrentModificationException
9. Exception

Iterator to overcome ConcurrentModificationException

If we try to remove an element while looping a collection

Inside the for or for-each loop we remove an element, it affects the boundary condition and hence we get this error.

Overcome this error, using ITERATOR instead of for or for-each loop

Iterator<Integer> It=marks.iterator();

while(it.hasNext())

{  
 Integer curr=it.next();

if(curr>50)

it.remove(); //no c m exception

}

Collections

List

Set

TreeMap Comparable Comparator

Map

Solution to overcome comparable/ comparator fear, solve req 4 or 5 in each mock.

SQL (mysql)

Handson challenges are added to /sql foder in git

String & Regex

String is immutable means we cannot modify the content of a String variable.

Regex can be useful in 2 situations:

1. Matches check if the string matches a pattern or not
2. Matcher extract group of word that matches a pattern

Task:

Solve java handson challenges now.

1. Exception

Abnormal events that occur during the execution of the program.

Every exception is defined with an exception class that is derived form

Throwable or (usually no)

Exception or (checked exception)

RuntimeException (unchecked exception)

Exception can be handled. Error should not be handled

Error is derived from Throwable

All Error sub classes are errors that are serious that the program must terminate

To throw an exception:

Create an object of Exception or any of its sub class and throw that object

Using throw keyword

Ex:

throw new Exception();

throw new ArithmeticException(“your message here”);

if you throw a checked exception inside a method,

then you have to declare that this method throws these exceptions using

“throws” clause. Or handle them using try-catch

public void method1()throws IOException, ParseExcepection, SQLException

instanceof

new

checked exceptions

InterruptedException

IOException

ParseException

1. <https://leetcode.com/>
2. <https://practice.geeksforgeeks.org/explore?page=1&sortBy=submissions&utm_source=geeksforgeeks&utm_medium=main_header&utm_campaign=practice_header>
3. <https://www.hackerrank.com/>
4. <https://www.codechef.com/>
5. <https://www.codewars.com/collections/java-basics>
6. <https://www.topcoder.com/>
7. <https://www.interviewbit.com/java-interview-questions/>

Remove method in collections:

**Question 6**

Marks: 1

**ArrayList to String Array**

Write a program that performs the following actions:  
  
1.Read m strings as input (fruit names).  
2.Create an arraylist to store the above m strings in this arraylist.  
3.Read n strings as input (fruit names).  
4.Create an arraylist to store the above n strings in this arraylist.  
5.Write a function fruitSelector which accepts the arraylists as input.  
6.Remove all fruits whose name ends with 'a' or 'e' from first arrayList and remove all fruits whose name begins  with 'm' or 'a' from second arrayList then combine the two lists and return the final output as a String array.  
7.If the array is empty the program will print as “No fruit found”  
Include a class UserMainCode with the static method **fruitSelector** which accepts the two arraylists and returns an array.  
  
Create a Class Main which would be used to read n strings and call the static method present in UserMainCode.  
  
**Input and Output Format:**  
  
Input consists of an integer (m) denoting the size of first arraylist. The next m elements would be the values of the first arraylist. The next input would be n denoting the size of the second arraylist. The next n elements would be the values of the second arraylist.  
  
Output consists of an array as per step 6. Refer sample output for formatting specifications.  
  
**Sample Input 1:**  
3  
Apple  
Cherry  
Grapes  
4  
Orange  
Mango  
Melon  
Apple  
**Sample Output 1:**  
Cherry  
Grapes  
Orange

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class UserMainCode3 {

static String[] fruitSelector(List<String> list1, List<String> list2) {

list1.removeIf((a)->a.toLowerCase().endsWith("a") || a.toLowerCase().endsWith("e"));

list2.removeIf((a)->a.toLowerCase().startsWith("m") || a.toLowerCase().startsWith("a"));

list1.addAll(list2);

String []array=new String[list1.size()];

return array;

}

}

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Main3 {

public static void main(String[] args) {

// TODO Auto-generated method stub

UserMainCode3 user3=new UserMainCode3();

Scanner sc=new Scanner(System.in);

int m=sc.nextInt();

List<String> list1=new ArrayList<>();

for(int i=0;i<m;i++)

{

String input=sc.nextLine();

if(input.equals(""))

input=sc.nextLine();

list1.add(input);

}

int n=sc.nextInt();

List<String> list2=new ArrayList<>();

for(int i=0;i<n;i++)

{

String input=sc.nextLine();

if(input.equals(""))

input=sc.nextLine();

list2.add(input);

}

String[] array=user3.fruitSelector(list1,list2);

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

{

System.out.println(array[i]);

}

}

}