| **Project** | **Status** | **Related files** | **Notes** |
| --- | --- | --- | --- |
|  | Not started | File |  |
|  | In progress | File |  |
|  | Launched | File |  |

Round 1 starts:-

OOPs Concept

Diff between Interface and abstract class

Interface example

1)Interface Payment{

  2) Public void DoPayment()   //payments;  // change the method name

3)}

4)Class PaymentService implements Payment{

5)Public void doPayment{

6) Return null;

7)}

}

How can i use payment service class in order service

Class OrderService{

@Autowired

PaymentService paymentservice;

}

List<Integer> nums={1,1,

1,0,0,0,1,1,1};

Sort this list without sort method or sort functionliaity

Give time complexity and space complexity

O/p={0,0,0,1,1,1,1,1,1}

Public List<Integer> sort(List<Integer> arr){

LIst<Integer> ans = new ArrayList<>();

Int countZero =0;

Int countOne = 0;

for(int i = 0; i < arr.size(); i++){

if(arr.get(i) == 0) countZero++;

else {

countOne++;

}

}

for(int i = 0; i <= countZero; i++){

ans.add(0);

}

for(int i = 0; i <= countOne; i++){

ans.add(1);

}

return ans;

}

Tc : O(N)

Sc :O(N)

2)

***Input:*** *arr[] = {2, 0, 2}*

*o/p-2*

***Input:*** *arr[]   = {3, 0, 2, 0, 4}*

*o/p->7*

*Public int MaxWater(int[] height){*

*Int n = height.length;*

*Int letfMax[] = new int[n];*

*Int rightMax[] = new int[n];*

*leftMax[0] = height[0];*

*Int ans = -1;*

*for(int i = 1 ; i < n*

*if(leftMax[i-1] < height[i]){*

*leftMax[i] = height[i];*

*}*

*else{*

*leftMax[i] = leftMax[i-1] ;*

*}*

*rightMax[n-1] = height[n-1];*

*for(int i = n-2; i>=0; i—){*

*if(rightMax[i+1] < height[i]){*

*rightMax[i] = height[i];*

*}*

*else{*

*rightMax[i] = rightMax[i+1];*

*}*

*}*

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

*Int maxHeight = Math.min(leftMax[i],rightMax[i]);*

*Int water = ( maxHeight -height[i]  ) an;*

*ans+=water;*

*}*

*return ans;*

*}*

*Question 3*

***Input:*** *strs = ["flower","flow","flight"]*

***Output:*** *"fl"*

*Can car cat*

*CA*

*Round -2*

*Public class House{*

*Public Void Kitchen(){*

*Do something;*

*}*

*Public Int Kitchin(int NoOfDrawoer){*

*Do something;*

*}*

*}*

*Public class OneBhk extends House{*

*@Override*

*Public Void Kitchen(){*

*Do something;*

*}*

*}*

*Public class OneBhkSpecialKitchen extends House{*

*@Override*

*Public int Kitcheni(int X){*

*Do something;*

*}*

*}*

*Public class print implement Runnable{*

*@Override*

*Run(){*

*for(int i = 1; i < 10; i++){*

*System.out.print(i);*

*}*

*}*

*Thread th = new Thread();*

*th.start();*

*}*

*Set*

*Map*

*List*

*Ram*

*ramShaym*

*Sorted array arr{1,4,5,8,11,12,15,18,19,25}*

*Public int find(int arr[],int K){*

*Int n = arr.length;*

*if(arr[n-1] < K){*

*return n-1;*

*}*

*Int ans = -1;*

*Int start = 0;*

*Int end = n-1;*

*while(start<end){*

*Int mid = (start + end)/2;    ///   1 2 3  5 6 7*

*if(arr[mid] == K){*

*return mid;*

*}*

*if(arr[mid] < K){*

*Start = mid +1;*

*Ans = mid;*

*}*

*else if(arr[mid] > K){*

*end = mid -1;*

*}*

*}*

*return ans;*

*}*

*Round 3 -*

*Q1) explain wait(), notify(), notifyAll(), sleep(), join()*

*!2) Thread T1 = 1,3,5,7,9...99*

*Thread T2 = 2,4,6,8,10..100*

*Write a program with the help of above two threads and make sure to get final outcome as 1,2,3,4,5,6,7,8,9...100*

*Public threadRun{*

*T1.start();*

*T1.Sleep(200);*

*T2.start();*

*T2.sleep(200);*

*}*

*Q2) Map<String,Integer> -> populate some values in the map*

*["first" : 3]*

*["second":1]*

*["third":2]*

*then sort the map based on values .*

*o/p = >.  ["second":1]  -> ["third":2] -> ["first" : 3]*

*Public Map<String,Integer>  sort(Map<String,Integer> map){*

*Int n = map.size();*

*Int arr[] = new int[n];*

*TreeMap<String,Integer> tree = new TreeMap<>();*

*for(x){*

*tree.put(x.getKey(),x.getValue());*

*}*

*return  tree;*

*Question 3 ) Design flight booking system like makeMyTrip*

1. *Let’s start with DB server  design ?*