ZEALOTS.JAVA SOURCE CODE

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.IOException;

import java.io.PrintWriter;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

public class Zealots {

private final int max;

private final int min;

private int turn;

public int count=0;

public static int num=0;

private final String name;

private final String record;

private final String turnf;

public Zealots(String a,int b,int c, int min, int max){

this.turn=c;

this.min=min;

this.max=max;

this.name=a+".txt";

this.record="Record "+a+".txt";

this.turnf="Turn "+a+".txt";

num = b;

}

public void turn(){

this.count++;

this.turn++;

num++;

try{

PrintWriter pw;

pw = new PrintWriter(new FileOutputStream(new File(this.name),true));

pw.printf("%03d",this.turn);

pw.println();

pw.close();

pw = new PrintWriter(new FileOutputStream(new File(this.record),true));

pw.printf("In: %03d - %s",this.turn,Date());

pw.println();

pw.close();

pw = new PrintWriter(new FileOutputStream(new File(this.turnf)));

pw.printf("%03d",this.turn);

pw.close();

pw = new PrintWriter(new FileOutputStream(new File("Total number in queue.txt")));

pw.printf("%d",num);

pw.close();

}

catch (IOException e) {

System.out.println("Problem with file output");

}

System.out.printf("Your number: %03d\n%s\n",this.turn,Date());

}

public String call(int n,int m) throws FileNotFoundException, IOException{

num--;

PrintWriter pw;

String s="0";

BufferedReader br = new BufferedReader(new FileReader(this.name));

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

s=br.readLine();

pw = new PrintWriter(new FileOutputStream(new File("Total number in queue.txt")));

pw.printf("%d",num);

pw.close();

pw = new PrintWriter(new FileOutputStream(new File(this.record),true));

pw.printf("Out: %s - %s",s,Date());

pw.println();

pw.close();

display(n,m);

return s;

}

public void display(int n,int m) throws FileNotFoundException, IOException{

PrintWriter pw;

String s="0";

BufferedReader br = new BufferedReader(new FileReader(this.name));

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

s=br.readLine();

pw = new PrintWriter(new FileOutputStream(new File("Display.txt"),true));

pw.printf("Number: "+s);

pw.println();

pw.printf("Counter:"+m);

pw.println();

pw.close();

}

private static String Date(){

DateFormat df = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

Date today = Calendar.getInstance().getTime();

return df.format(today);

}

}

MAININTERFACE.JAVA SOURCE CODE

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.IOException;

import java.io.PrintWriter;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.Scanner;

public class MainInterface {

static Scanner in = new Scanner(System.in);

static String[] args={};

static Zealots A,B,C;

static String h;

static int j,k;

static int turna, turnb, turnc;

static PrintWriter pw = null;

public static void main(String[] args) throws IOException {

BufferedReader br;

//read from number.txt to check the current volume of pending queue number

br = new BufferedReader(new FileReader("Total number in queue.txt"));

//br.readLine() return string, so assign it to h

h = br.readLine();

//if h is null or white space j=0, else convert h into integer value and assign to j

if (h==null)

j=0;

else

j=Integer.parseInt(h);

br = new BufferedReader(new FileReader("Turn A.txt"));

h = br.readLine();

if (h==null)

k=0;

else

k=Integer.parseInt(h);

//declaration for object A of class Service

A = new Zealots("A",j,k,0,400);

br = new BufferedReader(new FileReader("Turn B.txt"));

h = br.readLine();

if (h==null)

k=400;

else

k=Integer.parseInt(h);

//declaration for object B of class Service

B = new Zealots("B",j,k,400,600);

br = new BufferedReader(new FileReader("Turn C.txt"));

h = br.readLine();

if (h==null)

k=600;

else

k=Integer.parseInt(h);

//declaration for object B of class Service

C = new Zealots("C",j,k,600,999);

//read from file turn a/b/c to know the previous number that has been called by the hospital staff

br = new BufferedReader(new FileReader("Last turn - A.txt"));

h = br.readLine();

if(h==null) turna=0;

else turna=Integer.parseInt(h);

br = new BufferedReader(new FileReader("Last turn - B.txt"));

h = br.readLine();

if(h==null) turnb=0;

else turnb=Integer.parseInt(h);

br = new BufferedReader(new FileReader("Last turn - C.txt"));

h = br.readLine();

if(h==null) turnc=0;

else turnc=Integer.parseInt(h);

firstInterface();

}

public static void firstInterface() throws IOException{

System.out.println("1- number dispenser\n"

+ "2- counter\n"

+ "3- display");

System.out.println("menu? ");

int n=in.nextInt();

switch(n){

case 1: menu1();

break;

case 2: menu2();

break;

case 3: menu3();

break;

default: System.exit(0);

}

}

private static void menu1() throws IOException{

int n;

System.out.println("1- service A - General Enquiry\n"

+ "2- service B - Technical Assistance\n"

+ "3- service C - Billing/Payment\n"

+ "4 - Exit\n"

+ "Choose service: ");

do{

if( A.num < 40){

n=in.nextInt();

switch(n){

case 1: A.turn();

menu1();

break;

case 2: B.turn();

menu1();

break;

case 3: C.turn();

menu1();

break;

default: firstInterface();

break;

}

}

else{

System.out.println("\nPlease wait for awhile. We are out of tickets\n");

break;

}

}while(A.num<41);

firstInterface();

}

private static void menu2() throws IOException{

if(A.num==0){

System.out.println("\nThere is no pending number to call.");

firstInterface();

}

System.out.println("1 - Counter 1\n"

+ "2 - Counter 2\n"

+ "3 - Counter 3\n"

+ "4 - Counter 4");

int n=in.nextInt();

switch(n){

case 1:

case 2:

counter(n, "A");

break;

case 3:

counter(n, "B");

break;

case 4:

counter(n, "C");

break;

default:

break;

}

firstInterface();

}

private static void counter(int x,String a) throws FileNotFoundException, IOException{

String s=null;

int n,m=0;

System.out.println("Counter "+x);

System.out.println("Call next number? Enter 1");

if(in.nextInt()==1)

s=nextNum(x);

else

menu2();

System.out.println("Want to call again?\n1- Yes 2- No\n");

n=in.nextInt();

while(n==1 && m<3){

m++;

repeatNum(x,s,a);

n=in.nextInt();

}

menu2();

}

private static String nextNum(int x) {

String str=null;

try{

switch(x){

case 1: case 2:

turna++;

pw = new PrintWriter(new FileOutputStream(new File("Last turn - A.txt")));

pw.printf("%03d",turna);

str=A.call(turna,x);

break;

case 3:

turnb++;

pw = new PrintWriter(new FileOutputStream(new File("Last turn - B.txt")));

pw.printf("%03d",turnb);

str=B.call(turnb,x);

break;

case 4:

turnc++;

pw = new PrintWriter(new FileOutputStream(new File("Last turn - C.txt")));

pw.printf("%03d",turnc);

str=C.call(turnc,x);

break;

}

}

catch (FileNotFoundException e){

System.out.println("File not found !!");

}

catch (IOException e){

}

pw.println();

pw.close();

return str;

}

private static void repeatNum(int x,String str, String a) {

try{

pw = new PrintWriter(new FileOutputStream("Record "+a+".txt",true));

pw.printf("Call again: %s - %s",str,Date());

pw.println();

pw.close();

switch (x) {

case 1:

case 2:

A.display(turna, x);

break;

case 3:

B.display(turnb, x);

break;

default:

C.display(turnc, x);

break;

}

}

catch(IOException ex){

}

}

private static void menu3() {

try{

BufferedReader br = new BufferedReader(new FileReader("Display.txt"));

int lines=lines(),i;

System.out.println("lines: "+lines);

if(lines>8){

for(i=0; i<lines; i++){

if(i>lines-8)

System.out.println(br.readLine());

else

br.readLine();

}

}

else{

for(i=0; i<lines; i++)

System.out.println(br.readLine());

}

}

catch(FileNotFoundException ex){

System.out.println("File is not found !!!");

}

catch(IOException ex){}

}

private static int lines() throws FileNotFoundException, IOException{

BufferedReader br = new BufferedReader(new FileReader("Display.txt"));

int lines = 0;

while (br.readLine() != null)

lines++;

br.close();

return lines;

}

private static String Date(){

DateFormat df = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

Date today = Calendar.getInstance().getTime();

return df.format(today);

}

}