**VOTER (SUBJECT) 🡪 (OBSERVERED BY POLL)**

**POLL (SUBJECT AND OBSERVER) 🡪 (OBSERVERS VOTER AND IS A SUBJECT FOR SUBSCRIBERS)**

**SUBSCRIBERS (OBSERVERS) 🡪 (MEDIA WHO OBSERVERS POLLS)**

**MainClass.java**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Map;

import java.util.Scanner;

public class MainClass {

static String[] teams;

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

teams = new String[25];

AssociatedPress ap = new AssociatedPress();

USAToday ut = new USAToday();

LATimes lat = new LATimes(ap);

WashingtonPost wp = new WashingtonPost(ap);

SportsWeekly sw = new SportsWeekly(ap);

EspnToday et = new EspnToday(ap);

LATimes lat1 = new LATimes(ut);

WashingtonPost wp1 = new WashingtonPost(ut);

SportsWeekly sw1 = new SportsWeekly(ut);

EspnToday et1 = new EspnToday(ut);

Scanner sc = new Scanner(new File("./teams.txt"));

int counter = 0;

while (sc.hasNext()) {

teams[counter] = sc.next();

counter++;

}

ap.setTeams(teams);

ut.setTeams(teams);

for(int i = 1 ; i <= 100 ; i ++) {

VoterClass v = new VoterClass();

if(i <= 40) {

ap.subscribeToVoter(v);

}

if(i > 40 && i <= 60) {

ap.subscribeToVoter(v);

ut.subscribeToVoter(v);

}

if(i > 60) {

ut.subscribeToVoter(v);

}

v.votePolls();

}

ap.calculatePoll();

ut.calculatePoll();

}

}

**VoterSubject.java**

public interface VoterSubject {

public void registerPoll(PollSubject sb);

public void votePolls();

}

**VoterClass.java**

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Map;

import java.util.Random;

public class VoterClass implements VoterSubject{

ArrayList<PollSubject> polls;

private Map<String, Integer> teams;

VoterClass(){

polls = new ArrayList<PollSubject>();

}

@Override

public void registerPoll(PollSubject sb) {

// TODO Auto-generated method stub

polls.add(sb);

}

@Override

public void votePolls() {

// TODO Auto-generated method stub

Iterator<PollSubject> iterator = this.polls.iterator();

while(iterator.hasNext()) {

PollSubject poll = iterator.next();

teams = poll.getTeams();

Random r = new Random();

int currentRank = r.nextInt(26) + 1;

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

teams.put(key, currentRank);

while(teams.containsValue(currentRank)) {

currentRank = r.nextInt(26) + 1 ;

}

}

poll.getVote(teams);

}

}

}

**PollSubjectAndObserver.java**

import java.util.Map;

public interface PollSubjectAndObsever {

public void registerSubscribers(Subscribers sb);

public void notifySubscribers();

public Map<String, Integer> getTeams();

public void getVote(Map<String, Integer> teams);

}

**USAToday.java**

import java.util.ArrayList;

import java.util.Collections;

import java.util.HashMap;

import java.util.Iterator;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

public class USAToday implements PollSubjectAndObsever{

public Map<String, Integer> teams;

public ArrayList<Map> table;

ArrayList<Subscribers> subscribers;

USAToday(){

subscribers = new ArrayList<Subscribers>();

table = new ArrayList<Map>();

}

public void subscribeToVoter(VoterClass voter) {

voter.registerPoll(this);

}

@Override

public void registerSubscribers(Subscribers sb) {

// TODO Auto-generated method stub

subscribers.add(sb);

}

public void setTeams(String[] arr) {

teams = new HashMap<String, Integer>();

int counter = 1;

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

teams.put(arr[i], 0);

counter ++;

}

}

public Map<String, Integer> getTeams() {

Map<String, Integer> voteTeams = new HashMap<String, Integer>();

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

voteTeams.put(key, 0);

}

return voteTeams;

}

@Override

public void notifySubscribers() {

// TODO Auto-generated method stub

String[] arr = new String[teams.size()];

int counter = 0;

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

arr[counter] = key;

counter++;

}

Iterator<Subscribers> iterator = this.subscribers.iterator();

while(iterator.hasNext()) {

iterator.next().updateDisplay(arr,"ut");

}

}

public void calculatePoll() {

for (Map<String, Integer> entry : this.table) {

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

Integer value = entry.get(key);

teams.put(key, teams.get(key) + 26 - entry.get(key));

}

}

teams = sortByValue(teams);

System.out.println("\_\_\_\_\_\_TOTAL WEEKLY POINTS USAToday \_\_\_\_\_\_\_\_\_\_\_");

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

Integer value = teams.get(key);

System.out.println(key + " " + value);

}

notifySubscribers();

}

public void getVote(Map<String, Integer> vote){

this.table.add(vote);

}

public static <String, Integer extends Comparable<? super Integer>> Map<String, Integer> sortByValue(Map<String, Integer> map) {

List<Entry<String, Integer>> list = new ArrayList<>(map.entrySet());

list.sort(Collections.reverseOrder(Map.Entry.comparingByValue()));

Map<String, Integer> result = new LinkedHashMap<>();

for (Entry<String, Integer> entry : list) {

result.put(entry.getKey(), entry.getValue());

}

return result;

}

}

**AssociatedPress.java**

import java.util.\*;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.LinkedHashMap;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Random;

public class AssociatedPress implements PollSubjectAndObsever{

public Map<String, Integer> teams;

public ArrayList<Map> table;

ArrayList<Subscribers> subscribers;

AssociatedPress(){

table = new ArrayList<Map>();

subscribers = new ArrayList<Subscribers>();

}

public void subscribeToVoter(VoterClass voter) {

voter.registerPoll(this);

}

@Override

public void registerSubscribers(Subscribers sb) {

// TODO Auto-generated method stub

subscribers.add(sb);

}

public void setTeams(String[] arr) {

teams = new HashMap<String, Integer>();

int counter = 1;

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

teams.put(arr[i], 0);

counter ++;

}

}

public Map<String, Integer> getTeams() {

Map<String, Integer> voteTeams = new HashMap<String, Integer>();

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

voteTeams.put(key, 0);

}

return voteTeams;

}

@Override

public void notifySubscribers() {

// TODO Auto-generated method stub

String[] arr = new String[teams.size()];

int counter = 0;

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

arr[counter] = key;

counter++;

}

Iterator<Subscribers> iterator = this.subscribers.iterator();

while(iterator.hasNext()) {

iterator.next().updateDisplay(arr, "ap");

}

}

public void calculatePoll() {

for (Map<String, Integer> entry : this.table) {

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

Integer value = entry.get(key);

teams.put(key, teams.get(key) + 26 - entry.get(key));

}

}

teams = sortByValue(teams);

System.out.println("\_\_\_\_\_\_TOTAL WEEKLY POINTS AP\_\_\_\_\_\_\_\_\_\_\_");

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

Integer value = teams.get(key);

System.out.println(key + " " + value);

}

notifySubscribers();

}

public void getVote(Map<String, Integer> vote){

this.table.add(vote);

}

public static <String, Integer extends Comparable<? super Integer>> Map<String, Integer> sortByValue(Map<String, Integer> map) {

List<Entry<String, Integer>> list = new ArrayList<>(map.entrySet());

list.sort(Collections.reverseOrder(Map.Entry.comparingByValue()));

Map<String, Integer> result = new LinkedHashMap<>();

for (Entry<String, Integer> entry : list) {

result.put(entry.getKey(), entry.getValue());

}

return result;

}

}

**Subscribers.java**

public interface Subscribers {

public void updateDisplay(String[] apRanks, String pollName);

}

**LATimes.java**

import java.util.ArrayList;

public class LATimes implements Subscribers{

private ArrayList<PollSubject> polls;

private String[] pollRanks;

LATimes(PollSubject sub){

sub.registerSubscribers(this);

}

@Override

public void updateDisplay(String[] ranks, String pollName) {

// TODO Auto-generated method stub

this.pollRanks = ranks;

if(pollName.equals("ap")) {

DisplayScreenAP();

}else {

DisplayScreenUT();

}

}

public void DisplayScreenAP() {

System.out.println(" ");

System.out.println("\t\t LA Times");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Associated Press Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

public void DisplayScreenUT() {

System.out.println("\t\t LA Times");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("USA Today Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

}

**EspnToday.java**

import java.util.ArrayList;

public class EspnToday implements Subscribers{

private ArrayList<PollSubject> polls;

private String[] pollRanks;

EspnToday(PollSubject sub){

sub.registerSubscribers(this);

}

@Override

public void updateDisplay(String[] ranks, String pollName) {

// TODO Auto-generated method stub

this.pollRanks = ranks;

if(pollName.equals("ap")) {

DisplayScreenAP();

}else {

DisplayScreenUT();

}

}

public void DisplayScreenAP() {

System.out.println(" ");

System.out.println("\t\t ESPN Today");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Associated Press Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

public void DisplayScreenUT() {

System.out.println(" ");

System.out.println("\t\t ESPN Today");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("USA Today Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

}

**WashingtonPost.java**

import java.util.ArrayList;

public class WashingtonPost implements Subscribers{

private ArrayList<PollSubject> polls;

private String[] pollRanks;

WashingtonPost(PollSubject sub){

sub.registerSubscribers(this);

}

@Override

public void updateDisplay(String[] ranks, String pollName) {

// TODO Auto-generated method stub

this.pollRanks = ranks;

if(pollName.equals("ap")) {

DisplayScreenAP();

}else {

DisplayScreenUT();

}

}

public void DisplayScreenAP() {

System.out.println(" ");

System.out.println("\t\tWashington Post");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Associated Press Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

public void DisplayScreenUT() {

System.out.println(" ");

System.out.println("\t\tWashington Post");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("USA Today Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

}

**SportsWeekly.java**

import java.util.ArrayList;

public class SportsWeekly implements Subscribers{

private ArrayList<PollSubject> polls;

private String[] pollRanks;

SportsWeekly(PollSubject sub){

sub.registerSubscribers(this);

}

@Override

public void updateDisplay(String[] ranks, String pollName) {

// TODO Auto-generated method stub

this.pollRanks = ranks;

if(pollName.equals("ap")) {

DisplayScreenAP();

}else {

DisplayScreenUT();

}

}

public void DisplayScreenAP() {

System.out.println(" ");

System.out.println("\t\t Sports Weekly");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Associated Press Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

public void DisplayScreenUT() {

System.out.println(" ");

System.out.println("\t\t Sports Weekly");

System.out.println(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println(" USA Today Ranks");

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

System.out.println((i+1) + ". "+ pollRanks[i]);

}

}

}