

## Exemple que utilitza propietats de classe

Realitzarem el control d'una votació en la qual es pot presentar un número qualsevol de candidats. A cada moment es pot votar a qualsevol candidat i es poden demanar les següents dades:

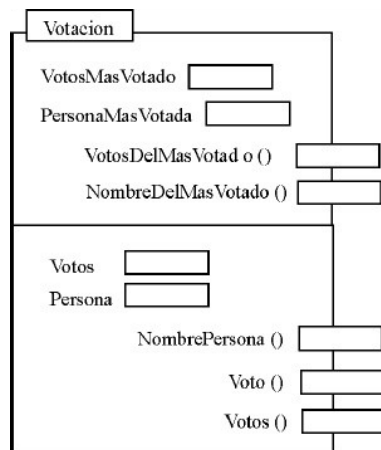
- Nom d'un candidat concret i el nombre de vots que porta fins al moment
- Nom del candidat més votat fins al moment i nombre de vots que porta aconseguits

La solució desenvolupada part d'una classe Votacion, que permet emmagatzemar el nom d'un candidat i el nombre de vots que porta, a més dels mètodes necessaris per a actualitzar l'estat de l'objecte. Si instanciamos la classe 14 vegades, per exemple, podrem portar el control de vots de 14 candidats.

La qüestió que ara se'ns planteja és:

### Com comptabilitzar el nombre de vots i emmagatzemar el nom del candidat més votat fins al moment?

- Una solució possible és crear una nova classe “MasVotado” que es instancie una sola vegada i continga propietats per a emmagatzemar aquests valors, al costat de mètodes per a consultar i actualitzar els mateixos. La solució exposada en el paràgraf anterior funcionaria correctament i no requereix de l'ús de propietats estàtiques, encara que existeix una qüestió de disseny que cal tindre clara: la classe “MasVotado” té sentit si s'instància una sola vegada?
- La solució que s'aporta en aquest apartat no requereix de l'ús d'una nova classe “MasVotado” o similar, ni necessita un número fix de instanciacions per a funcionar. La solució proposada conté les propietats i mètodes d'accés a la persona més votada dins de la pròpia classe Votacion, en la qual es vota a cada persona. Com el nom i nombre de vots de la persona més votada fins al moment és una informació general, que no depén únicament dels vots d'un candidat, sinó dels vots rebuts per tots els candidats, aquestes propietats han de ser accessibles, comunes i compartides per tots. Aquestes variables han de ser estàtiques (de classe).



## Codi del exemple

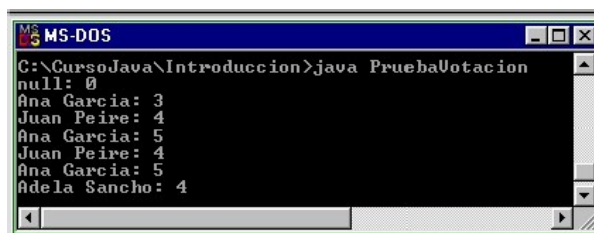
```
1      class Votacion {
2          // Persona a la que se vota en esta instancia y el numero
3          // de votos que lleva
4          private String Persona;
5          private int Votos;
6
7          // Persona mas votada de todas las instancias y el numero
8          // de votos que lleva
9          static private int VotosMasVotado = 0;
10         static private String PersonaMasVotada = null;
11
12         // Constructor
13         Votacion (String Persona) {
14             this.Persona = Persona;
15             this.Votos = 0;
16         }
17
18         // Se invoca cada vez que alguien vota a Persona
19         public void Voto() {
20             this.Votos++;
21             if (this.Votos > VotosMasVotado) {
22                 PersonaMasVotada = this.Persona;
23                 VotosMasVotado = this.Votos;
24             }
25         }
26
27         // Devuelve el nombre de Persona
28         public String NombrePersona() {
29             return this.Persona;
30         }
31
32         // Devuelve el numero de votos de Persona
33         public int Votos() {
34             return this.Votos;
35         }
36
37         // Devuelve el nombre de la persona mas votada
38         static public String NombreDelMasVotado() {
39             return PersonaMasVotada;
40         }
41
42         // Devuelve el numero de votos de la persona mas votada
43         static public int VotosDelMasVotado() {
44             return VotosMasVotado;
45         }
46     }
```

Amb la classe Votacion implementada, podem passar a comprovar el seu funcionament en una votació simulada de tres candidats: Juan, Ana i Adela. La classe PruebaVotacion realitza aquesta funció:

```
1      class PruebaVotacion {
2          public static void main (String[] args) {
3
4              System.out.println(Votacion.NombreDelMasVotado() + ": " + Votacion.VotosDelMasVotado());
5
6
7              // Tenemos tres candidatos en esta votación
8              Votacion Juan = new Votacion ("Juan Prado");
9              Votacion Ana = new Votacion ("Ana Garcia");
10             Votacion Adela = new Votacion ("Adela Sancho");
11
12             // empieza la votación
13             Juan.Voto(); Ana.Voto(); Ana.Voto(); Ana.Voto();Adela.Voto();
14             System.out.println (Votacion.NombreDelMasVotado() + ": " + Votacion.VotosDelMasVotado());
15
16
17             Juan.Voto(); Juan.Voto(); Juan.Voto(); Adela.Voto();
18             System.out.println (Votacion.NombreDelMasVotado() + ": " + Votacion.VotosDelMasVotado());
19
20
21             Adela.Voto(); Adela.Voto(); Ana.Voto(); Ana.Voto();
22             System.out.println (Votacion.NombreDelMasVotado() + ": " + Votacion.VotosDelMasVotado());
23
24
25             System.out.println (Juan.NombrePersona() + ": " +Juan.Votos() );
26             System.out.println (Ana.NombrePersona() + ": " +Ana.Votos() );
27             System.out.println (Adela.NombrePersona() + ": " +Adela.Votos() );
28
29         }
30     }
```

En la línia 4 s'accedeix al nom i nombre de vots del candidat més votat (i encara no hem creat cap candidat). Això és possible perquè tant els mètodes invocats com les variables accedides són estàtics, i tenen existència abans de la creació d'instàncies de la classe. Observe's com s'accedeix als mètodes a través del nom de la classe, no a través del nom de cap instància de la mateixa (que ni tan sols existeix en aquest moment). El valor esperat imprès per la instrucció System.out és el d'inicialització de les variables estàtiques en la classe Votacion (null i 0).

## Resultats



```
MS-DOS
C:\CursoJava\Introduccion>java PruebaVotacion
null: 0
Ana Garcia: 3
Juan Peire: 4
Ana Garcia: 5
Juan Peire: 4
Ana Garcia: 5
Adela Sancho: 4
```