

# Programación Orientada a Objetos. Práctica Adicional 1.x1

## Tema 1. Introducción a la Programación Orientada a Objetos

### Ejercicio 1.

1. Diseña la clase `Urna` que represente objetos *urnas* que permitan depositar votos (*afirmativos* o *negativos*), y permitan consultar el resultado de la votación (si la cuenta de votos afirmativos es mayor o igual que la cuenta de votos negativos).

La clase ofrecerá los siguientes constructores y métodos públicos:

- `Urna()`

Construye un objeto urna con las cuentas de votos afirmativos y negativos a cero.

- `void votar(boolean)`

Aumenta la cuenta de votos afirmativos o negativos según el valor recibido como parámetro.

- `boolean resultado()`

Devuelve el resultado de la votación (si la cuenta de votos afirmativos es mayor o igual que la cuenta de votos negativos).

2. Diseña la clase distinguida `PruebaUrna` (principal) que permita probar el funcionamiento de la clase `Urna`. Para ello, debe crear un objeto urna, depositar 3 votos afirmativos, depositar 2 votos negativos, mostrar un resultado intermedio de la votación, y volver a depositar 2 votos negativos. Finalmente mostrará el resultado final de la votación.

### Ejercicio 2.

1. Diseña la clase `SalaVotacion` que represente una sala de votación con dos *urnas*, que permita depositar votos (*afirmativos* o *negativos*) en cada urna, y permita consultar el resultado de la votación (si la cuenta de votos afirmativos es mayor o igual que la cuenta de votos negativos) en cada urna.

La clase ofrecerá los siguientes constructores y métodos públicos:

- `SalaVotacion()`

Construye un objeto sala de votación con dos objetos urna con las cuentas de votos afirmativos y negativos a cero.

- `SalaVotacion(Urna, Urna)`

Construye un objeto sala de votación con dos objetos urna externos, recibidos como parámetros.

- `void votar(int, boolean)`

Aumenta la cuenta de votos afirmativos o negativos según el valor recibido como parámetro de la urna identificada en el primer parámetro.

- `boolean resultado(int)`

Devuelve el resultado de la votación (si la cuenta de votos afirmativos es mayor o igual que la cuenta de votos negativos) de la urna identificada en el parámetro.

2. Diseña la clase distinguida `PruebaSalaVotacion` (principal) que permita probar el funcionamiento de la clase `SalaVotacion`. Para ello, debe crear un objeto sala de votación, depositar un voto negativo y dos votos afirmativos en la primera urna. Posteriormente deberá depositar un voto

positivo y dos votos negativos en la segunda urna. Finalmente, mostrará el resultado final de la votación de cada urna.

Después de depositar el primer voto en cada urna, mostrará un resultado intermedio de la votación de cada urna.

### Ejercicio 3.

1. Diseña la clase `UrnaOpaca` que represente objetos *urnas opacas*, que **son urnas** (hereda de la clase `Urna`) que sólo permiten depositar votos mientras no se consulte el resultado, pero una vez que se consulta el resultado, ya no permiten depositar más votos.

La clase ofrecerá los siguientes constructores y métodos públicos:

- `UrnaOpaca()`

Construye un objeto urna opaca con las cuentas de votos afirmativos y negativos a cero, y el estado de votación *abierto*.

- `boolean estaAbierta()`

Devuelve el estado de votación de la urna, `true` si la votación está abierta, y `false` en caso contrario.

- `void cerrarVotacion()`

Cierra la urna, poniendo el estado de votación a `false`.

- `void votar(boolean) // @Redefinición`

Si la urna está cerrada, entonces desecha el voto recibido como parámetro. Sin embargo, si la urna está abierta, entonces aumenta la cuenta de votos afirmativos o negativos según el valor recibido como parámetro.

- `boolean resultado() // @Redefinición`

Cierra la urna. Además, devuelve el resultado de la votación (si la cuenta de votos afirmativos es mayor o igual que la cuenta de votos negativos).

2. Diseña la clase distinguida `PruebaUrnaOpaca` (principal) que permita probar el funcionamiento de las clases `Urna` y `UrnaOpaca`. Para ello:

- a) Debe crear un objeto *urna*, depositar 3 votos afirmativos, depositar 2 votos negativos, mostrar un resultado intermedio de la votación, y volver a depositar 2 votos negativos. Finalmente mostrará el resultado final de la votación.
- b) Debe crear un objeto *urna opaca*, depositar 3 votos afirmativos, depositar 2 votos negativos, mostrar un resultado intermedio de la votación, y volver a depositar 2 votos negativos. Finalmente mostrará el estado de la urna opaca y el resultado final de la votación.