

# **CASO PRÁCTICO 2**

# • TÍTULO: Programación orientada a objetos con Java

## SITUACIÓN

Tenemos que resolver los siguientes problemas para la empresa de programación para la que trabajamos.

### INSTRUCCIONES

El objetivo de este ejercicio es crear una clase **Urna** cuyos objetos pueden contener bolas *blancas* y *negras*, y que nos permita realizar unas operaciones básicas sobre la misma. El nombre del proyecto ha de ser urnas.

La clase tendrá un par de variables de instancia (*protected*), negras y blancas, en las que se almacenará el *número de bolas* de cada color.

La clase Urna dispondrá de un constructor que permita crear instancias de la clase con el número inicial de bolas blancas y negras pasados como parámetros.

Además, incluirá métodos para:

- Consultar el número total de bolas que tiene (totalBolas()).
- Extraer una bola aleatoriamente y saber su color (bola()). El color vendrá dado por un enumerado. Para extraer una bola aleatoriamente se ha de sumar el número de bolas blancas y negras y tomar un número aleatorio entre 1 y dicha suma. Si ese número es menor o igual que el número de bolas blancas supondremos que la bola que sale es blanca; en otro caso, que es negra. Utilizad la clase java.util.Random para la generación de números aleatorios.

```
static private Random r = new Random();
int aleatorio = r.nextInt(valor) + 1;
```



Introducir una bola de un color determinado (ponerBlanca()
y ponerNegra()).

Implementar una aplicación que cree una urna y realice con ella el siguiente experimento:

- Mientras quede más de una bola en la urna, sacar dos bolas.
- Si ambas son del mismo color introducir una bola blanca en la urna; si son de distinto color introducir una bola negra (se supone que disponemos de suficientes bolas de ambos colores fuera de la urna).
- Por último, cuando quede sólo una bola, sacarla y mostrar su color.

Repetir la aplicación anterior pero dando el número de bolas blancas y negras iniciales al ejecutar el programa. Analizar los resultados obtenidos sobre el color de la bola final dependiendo del número de bolas iniciales y de su color.

# -blancas: int -negras: int -negras: int -rnd: Random +bola(): ColorBola +ponerBlanca() +ponerNegra() +totalBolas(): int +Urna(i:int, i1:int)

# #EjUrna(): +main(string: String[])