

Folla 3.3. Atributos e métodos estáticos 2

1. Imos modificar o exercicio da clase Bombilla. Engade á clase Bombilla un atributo *nome*, e un atributo *privado* **potenciaTotal**, na que gardamos a potencia total que está empregando o alumeado. Crea tamén un método estático que devolva o valor desa **potenciaTotal**. Queremos que se vaia gardando nese atributo o total de watos que están empregando as bombillas en cada momento. Pensa como terás que modificar o exercicio anterior para que isto aconteza: terás que modificar varios dos métodos xa creados.

Proba que funciona coas seguintes probas, comprobando o valor de **potenciaTotal**:

- Crea **bombilla1** (10W de inicio), **bombilla2** (15W de inicio), **bombilla3** (6W de inicio).
 - Mostra o valor de **potenciaTotal**.
 - Aumenta **bombilla1** en 5 W, **bombilla2** en 50 W.
 - Mostra a potencia de cada bombilla e o valor de **potenciaTotal**.
 - Pide por teclado un valor de watos por teclado para baixar bombilla2, e outro valor de watos por teclado para baixar bombilla3. Baixa a potencia das bombillas con eses valores.
 - Mostra a potencia de cada bombilla e o valor de **potenciaTotal**.
 - Baixa o valor das 3 bombillas para que chegue a 0, e comproba que valen 0 as 3 potencias e **potenciaTotal**.
 - Da un saúdo para indicar que o programa rematou.
2. Completa o exercicio anterior dando un nome a cada bombilla, para identificalas. Crea a continuación un método estático **comparaBombillas()** que compare a potencia de 2 bombillas, devolvendo o nome da bombilla con máis potencia de cada 2.

Intercala no **AppBombilla** unha chamada a ese método, despois da opción c) , f) e g).

3. Crea unha clase **Cilindro**, cun atributos radio e altura. A clase debe contar con 2 construtores, os setters e getters, e dous métodos **estáticos**, que reciban como parámetro o radio (rad) e altura (alt), e farán o cálculo da área e volume do cilindro. Emprega unha variable final para PI.

A área do cilindro é: $2 * PI * radio * (radio + altura)$, e o volume: $PI * radio^2 * altura$

Define tamén dous métodos non estáticos **calculaArea()**, e **calculaVolume()**, que deben calcular área e volume empregando os valores dos atributos do obxecto (deberán chamar aos métodos estáticos xa creados).

Crea na túa clase **AppCilindro** 3 cilindros e comproba todos os métodos da clase. Comproba que podes chamar aos métodos estáticos desde a clase Cilindro directamente.

4. Crea unha clase **Parcela**, cun atributos privados *nome*, *base* e *altura*. Define os seus getters e setters. Define tamén un método **área**, que calcule a área do rectángulo ($base * altura$). Crea agora un método **estático calculaMaiorArea2()**, que teña como argumentos 2 parcelas, e devolva o obxecto parcela con área maior. Define tamén outro método estático **estático calculaMaiorArea3()**, que faga o mesmo.

Crea nunha clase **AppParcela**, 5 parcelas, comproba que funcionan todos os métodos, e mostra finalmente os datos da parcela con maior área, empregando só **3** chamadas aos métodos creados.