

**Folla 3.3b. Atributos e métodos estáticos 2b**

1. Crea unha clase **Empleado** que terá como atributos o nome e a antigüidade do mesmo, ademais dun atributo estático que será o **salarioBase** dun empregado. A clase debe conter un método estático que permitirá aumentar o salario base na porcentaxe indicada e un método de instancia que fará o cálculo e devolverá o salario do empregado, que será o salario base máis 10€ (constante da clase) por cada ano de antigüidade.

Crea un programa **AppEmpleado** para probar a funcionalidade da clase. Crea tamén un método **cambiarSalarioBase()**, que permita cambiar ese valor. O seguintes Empleados deberán ter un soldo acorde ao novo valor de **salarioBase**.

2. Crea unha clase **Coche** que conteña os atributos privados *marca*, *modelo*, *cabalosFiscais* e *matricula*. Aos tres primeiros daráselles valor no construtor, á matrícula, no momento da venda. Existirá un atributo de clase **numCochesVendidos**. A clase terá un método **venda()** que recibirá a matrícula do coche e o escribirá no seu atributo, ademais de incrementar en 1 o número de coches vendidos.

Ademais, a clase conterá un método estático **calculaImposto()** para calcular o Imposto sobre Vehículos de Tracción Mecánica (IVTM), ao que se lle pasarán o número de cabalos fiscais, así como outro método de instancia que calcula o IVTM pasándolle ao devandito método estático os cabalos fiscais almacenados no atributo do obxecto.

- Menos de 8 cabalos fiscais 22,20 €
- De 8 ata 11,99 cabalos fiscais 61,75 €
- De 12 ata 15,99 cabalos fiscais 133,95 €
- De 16 ata 19,99 cabalos fiscais 179,22 €
- De máis de 20 cabalos fiscais 224,00 €

Define os métodos *getters* e *setters* para todos atributos.

Crea un programa principal para probar toda a funcionalidade da clase, creando 5 coches diferentes que cubran todas as posibilidades.

3. **Obxectos e arrays.** Crea unha clase **BiciElect** que teña como atributos **nome**, **marca**, **autonomiaKms**. Define os seus getters e setters, e 2 construtores, un con parámetros e outro sen eles. Define un atributo estático **numBicis** na que gardaremos o número de bicicletas que se van creando. Define tamén un método **comparaAutonomia()**, que recibe 2 **BiciElect** e devolve a **BiciElect** con maior autonomía.

Crea agora outra clase **AppBici**, na que debes crear un array con 5 bicis. O programa deberá mostrar o menú seguinte:

- a) Introducir unha única bicicleta, na posición que corresponde (primeiro no 0, logo no 1, etc.). Debes controlar con **numBicis** en que posición hai que introducir a bici nova.
- b) Mostrar todos os elementos do array. (emprega **numBicis** como índice máximo do bucle)
- c) Cambiar os datos dunha bicicleta concreta. O usuario será preguntado polo índice do array que quere modificar e modificar esa bicicleta
- d) Mostra a bicicleta con máis autonomía.
- e) Mostra a bicicleta con menor autonomía
- f) Saír.