

Programació orientada a objectes (I)

Exercicis (V)

STATIC & FINAL

Els modificadors static i final són opcionals, poden utilitzar-se tant en atributs com en mètodes i es poden combinar tots dos:

- **static:** L'atribut o mètode pertany a la classe (no a l'objecte). Per això es pot utilitzar sense instància cap objecte, des del nom de la classe: `NomClasse.atribut` o `NomClasse.metode(...)`.

Com que el valor s'emmagatzema a la classe NO pren valors diferents a cada objecte (com sí que succeeix amb els atributs 'normals' no static). Per exemple:

- L'atribut `salariMinim` d'`Empleat` (ss comú per a tothom, pot canviar).
- Mètodes útils com `Arrays.fill(...)` o `String.valueOf(...)` que podem utilitzar directament des de les classes `Arrays` i `String` sense instanciar un objecte.

És important saber que **des d'un mètode static no es poden fer servir atributs ni mètodes no static**. Al contrari sí que és possible.

- **final:** Un atribut final no es pot modificar. Podeu tenir valors diferents en cada objecte, però se n'ha de fixar el valor en el constructor. Un mètode final no es pot redefinir en una subclasse heretada (veurem herència a la següent unitat). Per exemple:

- El DNI de la classe `Persona`. No el podeu canviar i cada objecte té el seu.
- La combinació de static i final combina les dues característiques. Per exemple:
 - L'atribut `Math.PI` és static i final. Pertany a classe i no pot canviar.

En aquest apartat has de modificar els programes de l'apartat anterior (o fes una còpia del projecte si ho prefereixes) **i fer els canvis indicats.**

Exercici D1 – Punt

Necessitem un mètode que ens permeti crear un objecte `Punt` amb coordenades aleatòries. Aquesta funcionalitat no depèn de cap objecte concret per tant, serà estàtica. Haureu de crear un nou `Punt` (utilitza el constructor) amb `x` e `y` entre -100 i 100, i després tornar-lo (amb `return`).

- **public static Punt creaPuntAleatori()**

Prova-ho al main per comprovar que funciona. Crea diversos punts aleatoris amb `Punt.creaPuntAleatori()` i imprimeix el valor per pantalla.

Exercici D2 – Persona

El DNI d'una persona no pot variar. Afegeix el modificador final a l'atribut dni i assegura't que es guarde el valor al constructor. Elimineu el mètode **setDNI(...)** (que de totes maneres ja no es podrà utilitzar perquè Java no vos deixarà modificar l'atribut dni).

La majoria d'edat a les 18 anys és un valor comú a totes les persones i no pot variar. Crea un nou atribut anomenat **majoriaEdat** que siga static i final. Hauràs d'inicialitzar-lo a 18 en la declaració. Utilitza-lo en el mètode que comprova si una persona és major d'edat.

Crea un mètode **static boolean validarDNI(String dni)** que torne *true* si dni és vàlid (té 8 números i una lletra). Si no, tornarà *false*. Utilitzeu-lo al constructor per comprovar el dni (si no és vàlid, mostra un missatge d'error i no guardeu els valors).

Realitza algunes proves al main per comprovar el funcionament dels canvis realitzats. També podeu utilitzar `Persona.validarDNI(...)` per exemple per comprovar si uns DNI introduïts per teclat són vàlids o no (sense necessitat de crear cap objecte).

Exercici D3 – Rectangle

Necessitem fer alguns canvis perquè totes les coordenades estiguen entre (0,0) i (100,100). Afegeix a la classe Rectangle dos atributs anomenats min i max. Aquests valors són comuns a tots els objectes i no poden variar. Pensa quins modificats necessites afegir a min i max.

Utilitza min i max al constructor i als setters per comprovar els valors (com de costum, si no són correctes mostra un missatge d'error i no apliques els canvis).

També necessitem un mètode no constructor per crear rectangles aleatoris. Implementa'l.

Realitza proves al main per comprovar-ne el funcionament.

Exercici D4 – Article

A Espanya hi ha tres tipus d'IVA segons el tipus de producte:

- L'IVA general (21%): per a la majoria de productes a la venda.
- L'IVA reduït (10%): hoteleria, transport, habitatge, etc.
- L'IVA super reduït (4%): aliments bàsics, llibres, medicaments, etc.

Aquests tres tipus d'IVA no poden variar i a cada article se li aplicarà un dels tres.

Raona quins canvis caldria fer a la classe Artículo i implementa'ls.