

# Programació orientada a objectes (I)

## Exercicis (III)

### AFEGINT MÈTODES ÚTILS

Una classe ben dissenyada hauria d'incloure mètodes que facen operacions amb la informació dels objectes. D'aquesta manera, la classe disposarà de funcionalitats útils tant per a nosaltres com per a altres programadors. Això és una pràctica habitual i molt recomanable.

Per exemple, la classe **Scanner** té mètodes com **getInt()**, **getDouble()**, **getline()**, etc. que algú ha programat i podrem utilitzar quan els necessitem sense haver de programar-los nosaltres ni preocupar-nos per com funcionen internament. El mateix passa amb la classe **String** (**charAt()**, **substring()**, **toCharArray()**, etc.), la classe **Math** (**random**, **min**, **max**, **abs**, etc.), la classe **Arrays**, etc.

Totes aquestes són classes que Java incorpora per defecte (hi ha milers...). Hi ha moltes altres que permeten crear interfícies gràfiques amb botons, treballar amb imatges, llegir i escriure en arxius, reproduir música, enviar o rebre dades a través de la xarxa, etc.

En aquesta unitat estem aprenent a dissenyar i programar les nostres pròpies classes, els fonaments de la Programació Orientada a Objectes, necessari en qualsevol projecte de programari de certa envergadura.

Recorda que els mètodes poden ser **private** (només es poden utilitzar des de dins de la classe) o **public** (poden utilitzar-se des de fora, formen part de la interfície de la classe).

**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

Afegeix a la classe Punt els mètodes públics següents:

- **public void imprimeix()** // Imprimeix per pantalla les coordenades. Exemple: "(7, -5)"
- **public void setXY(int x, int y)** // Modifica les dues coordenades. És com un setter doble.
- **public void desplaça(int distx, int disty)** // Desplaça el punt la quantitat (distx,disty) indicada. Exemple: Si el punt (1,1) es desplaça (2,5) llavors estarà a (3,6).

- **public int distancia(Punt p)** // Calcula i torna la distància entre el propi objecte (this) i un altre objecte (Punt p) que es passa com a paràmetre: [distància entre dues coordenades](#).

Proveu d'utilitzar aquests mètodes des del main per comprovar-ne el funcionament.

### Exercici D2 – Persona

Afegeix a la classe Persona els mètodes públics següents:

- **public void imprimeix()** // Imprimeix la informació de l'objecte: "DNI:... Nom:... etc."

- **public boolean esMajorEdat()** // Retorna true si és més gran (false si no).

- **public boolean esJubilat()** // Retorna true si té 65 anys o més (false si no).

- **public int diferenciaEdat(Persona p)** // Retorna la diferència d'edat entre 'this' i p.

Proveu d'utilitzar aquests mètodes des del main per comprovar-ne el funcionament.

### Exercici D3 – Rectangle

Afegeix a la classe Rectangle mètodes públics amb les funcionalitats següents:

- Mètode per **imprimir** la informació del rectangle per pantalla.

- Mètodes setters dobles i quadruples: **setX1Y1**, **set X2Y2** i **setAll**.

- Mètodes **getPerimetre** i **getArea** que calculen i tornen el perímetre i l'àrea de l'objecte.

Proveu d'utilitzar aquests mètodes des del main per comprovar-ne el funcionament.

### Exercici D4 – Article

Afegeix a la classe Article mètodes públics amb les següents funcionalitats:

- Mètode per **imprimir** la informació de l'article per pantalla.

- Mètode **getPVP** que retorne el preu de venda al públic (PVP) amb iva inclòs.

- Mètode **getPVPDescompte** que retorne el PVP amb un descompte passat com a argument.

- Mètode  **vendre**  que actualitza els atributs de l'objecte després de vendre una quantitat 'x' (si és possible). Tornarà *true* si ha estat possible (*false* en cas contrari).

- Mètode **emmagatzemar** que actualitza els atributs de l'objecte després d'emmagatzemar una quantitat "x" (si és possible). Tornarà *true* si ha estat possible (false en cas contrari).

Proveu d'utilitzar aquests mètodes des del main per comprovar-ne el funcionament.