

# Disseny i programació orientada a l'objecte

David García Solórzano

PID\_00235244

Material docent de la UOC

**David García Solórzano**

Graduat superior en Enginyeria en Multimèdia i enginyer informàtic per la Universitat Ramon Llull des dels anys 2007 i 2008, respectivament. És també doctor per la Universitat Oberta de Catalunya des del 2013, on va realitzar una tesi doctoral relacionada amb l'aprenentatge virtual. Des del 2008 és professor de la Universitat Oberta de Catalunya en els Estudis d'Informàtica, Multimèdia i Telecomunicació.

Primera edició: setembre 2016

© David García Solórzano

Tots els drets reservats

© d'aquesta edició, FUOC, 2016

Av. Tibidabo, 39-43, 08035 Barcelona

Disseny: Manel Andreu

Realització editorial: Oberta UOC Publishing, SL

Dipòsit legal: B-13.008-2016

*Cap part d'aquesta publicació, incloent-hi el disseny general i la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric com químic, mecànic, òptic, de gravació, de fotocòpia o per altres mètodes, sense l'autorització prèvia per escrit dels titulars del copyright.*

## Introducció

Aquesta assignatura introdueix el paradigma de la programació orientada a objectes (POO), que facilita la modelització d'entitats del món real a partir de les seves propietats i comportaments. Aquest paradigma es considera una evolució del paradigma de la programació estructurada après en l'assignatura anterior del grau. En aquest sentit, veurem, tant en el pla teòric com pràctic, els avantatges que la POO ofereix respecte a la programació estructurada.

El material docent de l'assignatura ens aproparà a la POO des d'un punt de vista teòric oferint-nos pinzellades d'aspectes pràctics que han de ser considerats. Els materials són estructurats en 5 mòduls que volen introduir de manera progressiva l'estudiant en aquest nou paradigma. La POO requereix un canvi en la manera de pensar i dissenyar els programes respecte al paradigma de la programació estructurada que l'estudiant ja coneix. Per això, aquesta assignatura exigeix, durant tot el curs i especialment al començament, una dedicació especial de l'estudiant.

El mòdul «Introducció al paradigma de la programació orientada a objectes» explica com es va passar del paradigma de la programació estructurada al paradigma de la programació orientada a objectes (POO). Així mateix, ofereix una visió global d'aquest paradigma, n'esmenta els conceptes bàsics, en destaca els beneficis i explica els reptes als quals un programador novell ha de fer front quan comença amb la POO.

El mòdul «Objecte i classe» se centra en els dos elements principals de la POO: l'objecte i la classe. Aquests dos elements són explicats, d'una banda, mitjançant exemples de la vida quotidiana i, de l'altra, amb coneixements adquirits en l'assignatura prèvia, com és el cas del mòdul i la tupla. En aquest mòdul es treballa la classe de manera individual, és a dir, sense relacionar-la amb altres classes. A més, s'introdueix el llenguatge de modelització de sistemes de programari UML (*Unified Modeling Language*), que permet representar una classe de manera formal i gràfica. Aquest llenguatge ens acompanyarà en la resta de mòduls. Per acabar, s'explica l'objecte com una instanciació de la classe i com els objectes es relacionen entre si mitjançant missatges.

El mòdul «Encapsulació i extensió» presenta un dels majors beneficis de la POO: l'encapsulació. Amb ella s'explica la reutilització de codi i l'ocultació dels membres d'una classe, és a dir, els atributs i els mètodes.

El mòdul «Associació i herència» fa un pas endavant i explica la relació entre instàncies –anomenada associació– i entre classes –coneguda com a relació de generalització/especialització, o més comunament, com a herència. Aquest mòdul, a més de l'explicació teòrica, mostra la transformació en codi font de cada tipus de relació.

El mòdul «Tipus de classe i interfície» fa un repàs per diferents tipus de classe que difereixen de la classe estàndard vista fins al moment. Així mateix, presenta el concepte d'interfície com un element similar a la classe que facilita el desenvolupament en equip i ajuda a jerarquitzar les classes del nostre programa.

Finalment, cal indicar que la part pràctica de l'assignatura –portada a terme durant l'avaluació contínua– ens permetrà experimentar amb la POO. Per a això utilitzarem el llenguatge de programació Java.

## Objectius

Els materials didàctics d'aquesta assignatura aportaran els coneixements necessaris per a tenir una visió clara del paradigma de la programació orientada a objectes i permetran assolir els objectius següents:

- 1.** Conèixer els conceptes bàsics de la programació orientada a objectes: objecte, classe, encapsulació, ocultació, associació, herència, interfície, etc.
- 2.** Identificar i descriure les classes que ens permetin representar i modelar els elements d'un problema i les relacions entre ells mitjançant diagrames de classe UML.
- 3.** Aprendre a programar en Java per mitjà dels recursos propis de la programació orientada a objectes.
- 4.** Adquirir destresa en la utilització de les eines de desenvolupament en Java.

## Continguts

Mòdul didàctic 1

### **Introducció al paradigma de la programació orientada a objectes**

David García Solórzano

1. Context: programació estructurada
2. I arriba la programació orientada a objectes
3. De la programació estructurada a la programació orientada a objectes
4. Beneficis de la programació orientada a objectes
5. Enfrontar-se a la programació orientada a objectes

Mòdul didàctic 2

### **Objecte i classe**

David García Solórzano

1. Definir els conceptes *objecte* i *classe*
2. Classe
3. Objecte

Mòdul didàctic 3

### **Encapsulació i extensió**

David García Solórzano

1. Què és l'encapsulació?
2. Reutilització de codi
3. Ocultació
4. Extensió d'atributs i mètodes d'una classe

Mòdul didàctic 4

### **Associació i herència**

David García Solórzano

1. Associacions: relació entre instàncies
2. Herència: relació de generalització/especialització o entre classes
3. Exemple resum

Mòdul didàctic 5

### **Tipus de classe i interfície**

David García Solórzano

1. Tipus de classe
2. Interfície