Introducció a la Programació Orientada a Objectes (POO)

Institut Rafael Campalans
Plaça del Remei, 1
17160 Anglès

Cicle: DAM
Curs: 2022/2023
Mòdul: 02 Bases de Dades

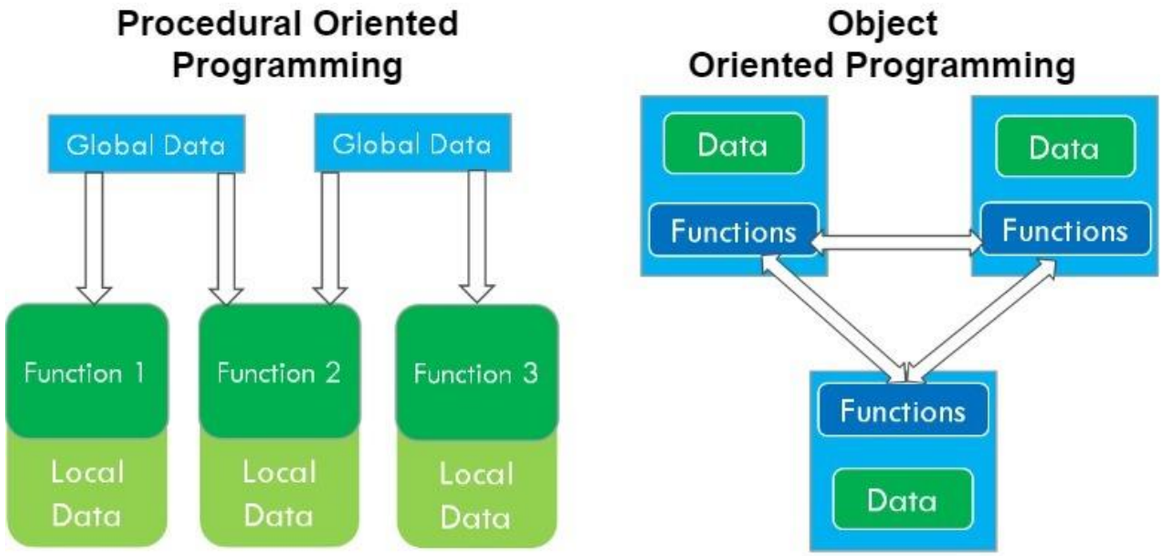
Objectius

- Introducció a la programació orientada a objectes
- Evolució dels mecanismes d'abstracció

Evolució dels mecanismes d'abstracció (1)

Abstracció:

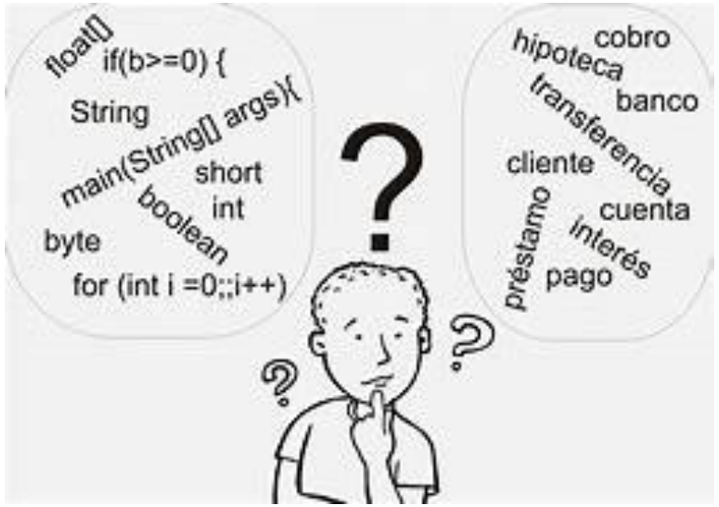
- capacitat per a encapsular i aïllar la informació
- és la principal eina per a reduir la "complexitat"
- fomenta la reutilització del codi



Evolució dels mecanismes d'abstracció (2)

Evolució:

1. Procediments i funcions
2. Mòduls (o unitats, o llibreries, o paquets)
3. Tipus de Dades Abstractes (TDAs)
4. Objectes



Procediments i funcions

- Permeten agrupar una sèrie d'instruccions en accions
- Tenim així una "ocultació del codi" que permet que un procediment sigui emprat per molts de programadors, sense que necessàriament sàpiguen els detalls de la seva implementació

Mòduls, unitats, llibreries o paquets

- Són una agrupació de procediments i funcions
- Es fa pública la seva definició (o prototipus), mentre que la implementació queda privada.
- Regles de David Parnas (1972):
  1. S'ha de proporcionar a l'usuari del mòdul tota la informació necessària per usar-lo correctament, i res més
  2. S'ha de proporcionar a l'implementador del mòdul tota la informació necessària per programar-lo correctament, i res més

Tipus de Dades Abstractes (TDAs)

- Ajunten l'estructura de dades amb las operacions possibles sobre aquesta estructura.

- Exemple: la llista dinàmica:
  - definició de node i llista
  - procediment crear_llista_buida
  - procediment inserir_node
  - ...

- Altres exemples: piles, coes, arbres, ...

Objectes

- També consisteix en encapsular l'estructura de dades (propietats o atributs de l'objecte) i les operacions sobre aquestes (mètodes)

- Incorpora alguns conceptes nous, encaminats a millorar l'abstracció i fomentar la reutilització del codi:

    - Pas de missatges
    - Herència
    - Polimorfisme

L’exemple de la florista (1)

- Vull enviar flors a la meva amiga (Margalida) que viu a Mallorca.
- Això és una tasca impossible per jo, però és fàcil fer-ho utilitzant una florista de Barcelona (Rosa).
- Només he de dir-li quina classe i quantitat de flors vull i l’adreça on s’han d’enviar i ja no m’he de preocupar de res més.

L’exemple de la florista (2)

Què he fet ?

- He cercat un agent apropiat (un objecte) i li he passat un missatge amb la meva petició.
- És ara responsabilitat de na Rosa satisfer la meva sol·licitud
- Per fer-ho, na Rosa empra un mètode, és a dir un conjunt d’operacions (un algorisme). Jo no tinc ni idea de com ho fa, no sé els detalls d’implementació d’aquest mètode.

L’exemple de la florista (3)

MISSATGES i MÈTODES

- En la POO l’acció s’inicia mitjançant la transmissió d’un missatge a un agent (un objecte), responsable de l’acció.
- El missatge porta codificada la petició d’una acció i s’acompanya de qualsevol informació addicional (arguments) necessària per dur a terme la petició.
- El receptor (l’objecte que rep el missatge), si accepta el missatge, executarà algun mètode per satisfer la petició i es responsabilitza de dur a terme l’acció.

L’exemple de la florista (4)

- En aquest procés hi ha el que es diu una ocultació d’informació (exemple de conduir un cotxe)

- Comparació missatge/crida a procediment:

Ambdós són una petició d’una acció, amb un conjunt d’arguments
Diferències:
el missatge té un receptor designat, mentre que el procediment no
la interpretació del missatge depèn del receptor i pot variar segons quin sigui el receptor, per exemple:
el meu dentista (Joan) no té cap mètode per enviar flors a la meva tia
la meva amiga Maria potser sí entén el missatge i té un altre mètode per dur-lo a terme

L’exemple de la florista (5)

Comparació missatge/crida a procediment:

- Resumint, en el pas de missatges hi ha un receptor designat i la interpretació (la tria del mètode que faci) pot diferir amb receptors diferents

- Normalment, el receptor específic només es coneix en temps d’execució (això es diu enllaç tardiu), mentre que en el cas dels procediments, l’enllaç es fa en temps de compilació o linkatge (enllaç primerenc)

L’exemple de la florista (6)

Per què jo li paso el missatge a Rosa (la florista) i no a Joan (el dentista) ?

Perquè sé coses dels floristes en general i espero que na Rosa, per ser un exemplar (una instància) d’aquesta categoria (florista) entrarà en aquest patró general.

Podem emprar el terme Florista per representar aquesta categoria (classe) de tots els floristes

L’exemple de la florista (7)

CLASSES i INSTÀNCIES (o exemplars)

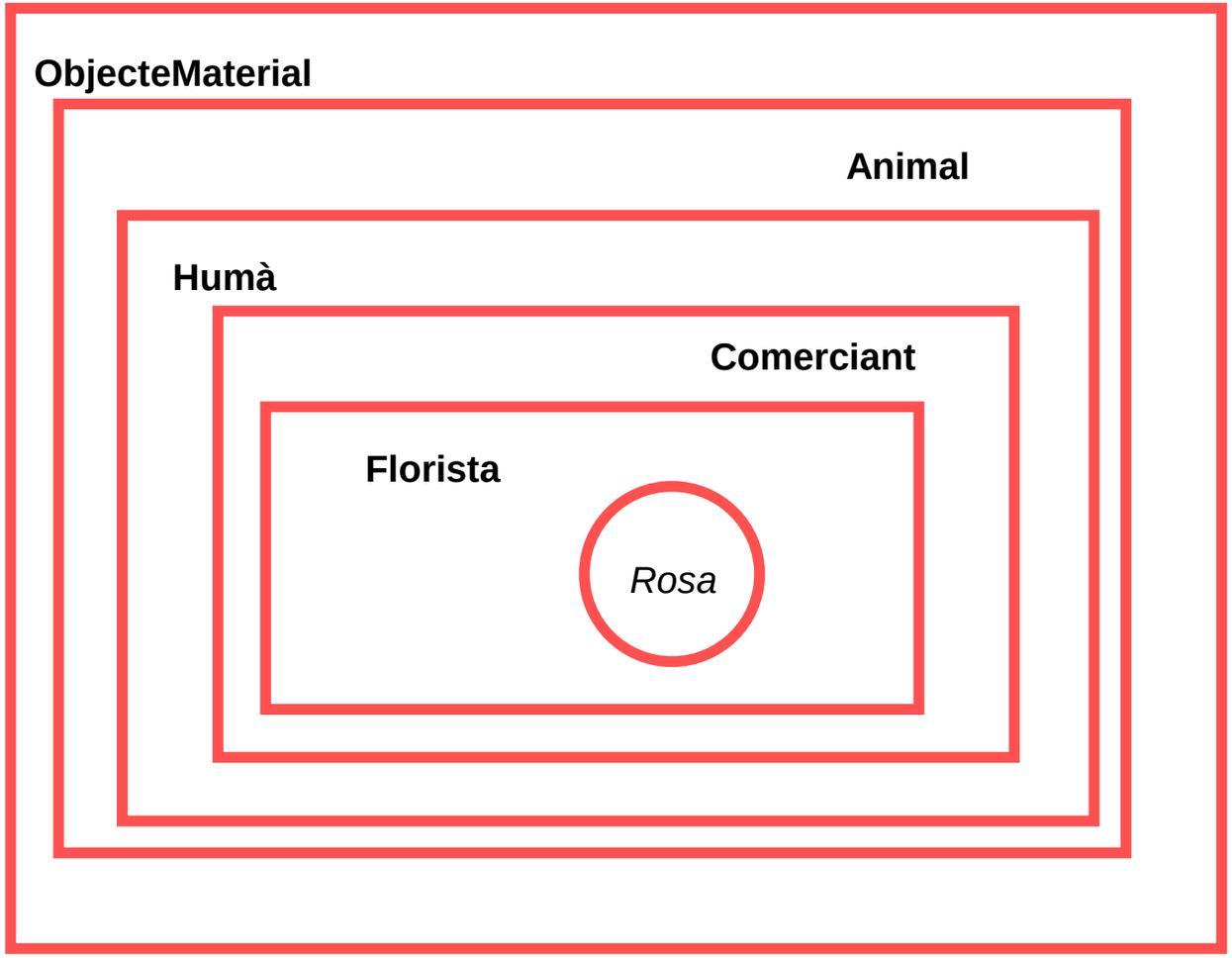
Amb una classe modelem les característiques d’un conjunt d’individus. La classe és una abstracció que ve definida per uns tipus de dades(atributs o propietats) i pels mètodes.
Tots els objectes són exemplars o instàncies d’una classe (regla de l’objecte). Una classe és un model i un objecte és una instància (un exemplar) d’aquest model
Tots els objectes d’una classe es diferencien entre ells perquè les seves propietats o atributs prenen valors diferents, però no obstant, utilitzaran el mateix mètode en resposta a missatges similars.
Així, el mètode invocat per un objecte en resposta a un missatge queda determinat per la classe del receptor
Allò important són las classes no els objectes: podríem dir millor Programació Orientada a Classes

L’exemple de la florista (8)

Apart, na Rosa per ser florista, també és comerciant.
La classe Florista és una forma especialitzada de la classe Comerciant
Un Comerciant és també un Humà
I un Humà és un Mamífer
I un Mamífer un Animal
I podem acabar dient que un Animal és un ObjecteMaterial (té massa i pes)

L’exemple de la florista (9)

- Jerarquia de classes:



L’exemple de la florista (10)

Tenim una jerarquia de classes. Per exemple, Humà és superclasse de Comerciant i Florista és subclasse de Comerciant

Herència:

- Cada nova subclasse inclou (hereta) tot el comportament i atributs dels seus antecessors.
- Així doncs, el comportament d’una classe més general és aplicable també a les classes més específiques.

L’exemple de la florista (11)

Herència múltiple:

- Una classe hereta propietats i mètodes no només d’una altra classe jeràrquicament superior, sinó de més d’una.
- Molts de llenguatges (Java entre ells) no la suporten

L’exemple de la florista (12)

Excepcions en l’herència:

- Pot passar que el fet de pertànyer a una classe anul·li algunes característiques de classes superiors.
- Per exemple, la classe Ornitorrinc és subclasse de Mamífer, però un ornitorrinc posa ous.
- Enllaç de mètodes:
  - Per cercar un mètode que es pugui invocar, primer es cerca de la classe receptor, si no hi és, en la seva superclasse i així successivament (no a l’enrevés, ja que poden haver mètodes anul·lats)

L’exemple de la florista (i 13)

Polimorfisme:

- Literalment significa un objecte i moltes formes
- Per herència:
  - Per exemple, la meva amiga Amapola i la florista Rosa responen de manera diferent front el mateix missatge, (els seus mètode “enviar flors” seran diferents)
- Per sobre-càrrega:
  - És un concepte que permet que un mètode tingui múltiples implementacions que es seleccionaran en base a quin tipus se li passa en el missatge. Per exemple, la classe Ca i el mètode “ensumar”. El comportament del ca serà distint si el que ensuma és un moix o és menjar.
  - La sobrecàrrega és estàtica, l’implementador ha de saber tots els tipus que es pot trobar com a paràmetre

A partir d’ara ...

Tots els llenguatges moderns tenen programació orientada a objectes (c#,Python,Java) i alguns d’ells es nadiu (tot son objectes).

A partir d’ara centrarem la programació orientada a objectes amb bases de dades i en el nostre cas amb Postgresql.

WEBGRAFIA

- Programación orientada a objetos, Wikipedia, Setembre 2022, https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos