

Programación orientada a objetos

Ejemplos Tema 1: Conceptos OO

[De T. Budd, *Introducción a la Programación Orientada a Objetos*, cap. 1]

Descripción de la POO según Alan Kay (*creador de Smalltalk, a principios de los '70*)

La POO está basada en el principio de diseño recursivo:

1. Todo son objetos.
2. Los objetos funcionan haciéndose peticiones de uno a otro mediante paso de mensajes.
3. Cada objeto tiene su propia memoria, que consta de otros objetos.
4. Cada objeto es un ejemplar (*instancia*) de una clase. Una clase agrupa objetos similares.
5. La clase es el repositorio del comportamiento asociado con un objeto.
6. Las clases están organizadas en una estructura arbórea que se denomina jerarquía de herencia.

Veamos cómo aparecen estos conceptos al solucionar un problema de la vida real.

Ejemplo de conceptos de OO: Enviar flores a una amiga

Considérese el problema de enviar flores a una amiga que reside en una ciudad diferente. Por ejemplo, Juan quiere enviar flores a Inés.

Juan no puede enviarlas directamente y para ello utiliza los servicios de su florista local, asociado a Interflora.

Juan le indica al florista (que se llama Paco) la dirección de Inés, cuánto quiere gastar, el tipo de flores que enviar y un texto para poner en la tarjeta que acompaña al ramo.

Paco contacta con una floristería en la ciudad de Inés, que se encarga de preparar el ramo de flores y buscar el repartidor que lo entregue.

Si se piensa con detenimiento, puede haber además otras personas involucradas en esta transacción. Por ejemplo, el mayorista de flores, tal vez alguien que prepare el ramo, etc.

Agentes y comunidades

Lo primero que se puede ver es que los resultados se consiguen por la interacción de unos agentes, que llamamos objetos.

Además, cualquier actividad no trivial requiere la interacción de una comunidad de objetos que trabajan juntos.

Cada objeto tiene una parte que jugar, un servicio que proporcionar a los otros miembros de la comunidad.

Elementos de la POO - Objetos

Según el primer principio de Kay:

1. Todo son objetos.

Las acciones en POO las realizan los objetos.

Normalmente hay muchos objetos que colaboran en un escenario. Por ejemplo, D. Juan, el florista, el que prepara el ramo, etc.

Cada objeto tiene una responsabilidad, y el resultado es el producto de la colaboración de todos los objetos en la solución del problema.

Elementos de la POO – Mensajes

El principio número 2:

2. Los objetos funcionan haciéndose peticiones de uno a otro mediante paso de mensajes.

Las acciones en POO se producen como respuesta a las peticiones de acciones, que se llaman mensajes.

Un objeto puede aceptar un mensaje y como respuesta realizará una acción y devolverá un valor.

Para empezar el proceso de enviar flores, Juan le da un mensaje al florista, que a su vez envía un mensaje al florista en la localidad de la destinataria, que le da un mensaje al repartidor, y así sucesivamente.

Ocultando información

Obsérvese que como usuario de un servicio que proporciona un objeto lo único que se necesita saber es el nombre de los mensajes que aceptará el objeto.

No hace falta tener ninguna idea sobre cómo se realizarán las acciones en respuesta a la petición.

Una vez aceptado un mensaje, un objeto es responsable de llevarlo a cabo.

Elementos de la POO – Receptores

Los mensajes son diferentes de las llamadas a función tradicionales en dos aspectos importantes:

- En un mensaje hay designado un receptor que acepta el mensaje.
- La interpretación del mensaje puede ser diferente, dependiendo del receptor.

Diferentes receptores, mismo mensaje, diferentes acciones

var

Paco : Florista;

```
Carmen : Hermana;
Sacamueles : Dentista;

begin
    Paco.enviaFloresA(miNovia); { funcionará }
    Carmen. enviaFloresA(miNovia); { también funcionará }
    Ken. enviaFloresA(miNovia); { probablemente no funcionará }
end;
```

El mismo mensaje ocasionará acciones distintas, dependiendo de a quién vaya dirigido.

Comportamiento e interpretación

Aunque varios objetos diferentes puedan aceptar el mismo mensaje, las acciones (*comportamiento*) que realizará el objeto podrán ser diferentes.

La determinación de qué comportamiento realizará el objeto se toma en tiempo de ejecución (vinculación tardía, o *late binding*).

El hecho de que el mismo nombre pueda significar dos operaciones completamente diferentes es una forma de *polimorfismo*.

Elementos de la POO – Diseño Recursivo

3. Cada objeto tiene su propia memoria, que consta de otros objetos.

Cada objeto es como un computador en miniatura, un procesador especializado en realizar una tarea específica.

Elementos de la POO – Clases

4. Cada objeto es un ejemplar (*instancia*) de una clase. Una clase agrupa objetos similares.
5. La clase es el repositorio del comportamiento asociado con un objeto.

El comportamiento I que se espera de Paco está determinado por la idea general que yo pueda tener del comportamiento de los floristas.

Se dice que Paco es una *instancia* de la *clase* Florista.

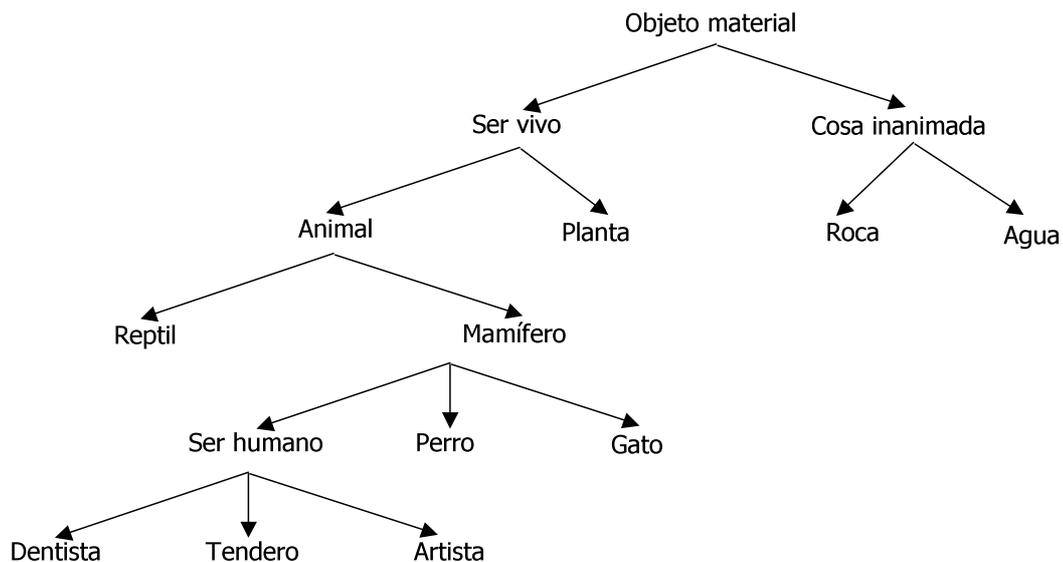
El comportamiento está asociado con las clases, no con instancias individuales. Todos los objetos que son instancias de una clase utilizan el mismo método en respuesta a mensajes similares.

Jerarquías de categorías

Pero hay más cosas que puedo saber sobre Paco, además de que es un Florista. También es un Tendero, y un Humano, y un Mamífero, y muchas más cosas.

En cada nivel de abstracción podemos registrar cierta información. Esa información será aplicable a todos los niveles inferiores, más especializados.

Jerarquías de clases



Elementos de la POO – Herencia

6. Las clases están organizadas en una estructura arbórea que se denomina jerarquía de herencia.

La información (datos y comportamiento) que le asociamos a un nivel de abstracción en la jerarquía de clases es automáticamente aplicable a los niveles inferiores de la jerarquía.

Elementos de la POO – Sobreescritura

Las subclasses pueden alterar o sobrescribir información heredada de las clases padre:

Todos los mamíferos dan de mamar a los recién nacidos.

La herencia combinada con la sobreescritura es una de las características más potentes de la POO.

Computación como simulación

La visión de la computación en POO es similar a crear un universo de objetos computacionales interactuando entre sí.

En definitiva, la POO es similar a la manera en que la gente piensa cómo resolver problemas en la vida real: que sea otro agente quien haga el trabajo real.