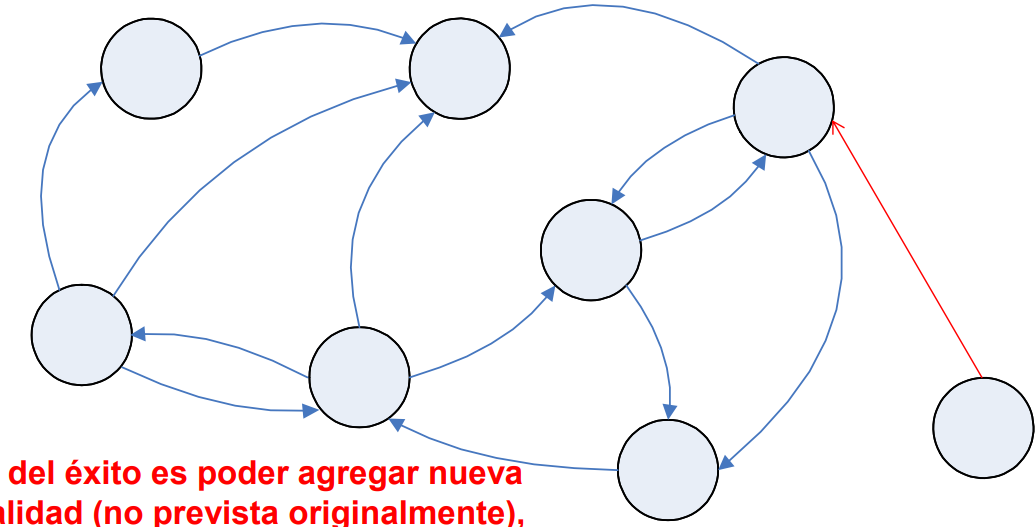
Orientacion a Objetos 1:

Teoria 1 – 29/8/24:

Software construido con objetos:

Es un conjunto de objetos que colaboran enviándose mensajes. Todo computo ocurre dentro de los objetos.



La clave del éxito es poder agregar nueva funcionalidad (no prevista originalmente), reemplazar objetos o modificar objetos y que el sistema “no se entere”, ni se rompa. E.g. integración Whatsapp y messenger Facebook

Los sistemas están compuestos (solamente) por un conjunto deobjetos que colaboran para llevar a cabo sus responsabilidades

Los objetos son responsables de:

* conocer sus propiedades
* conocer otros objetos (con los que colaboran)
* llevar a cabo ciertas acciones.

Aspectos de interés:

* No hay un objeto main.
* Cuando codificamos, describimos (programamos) clases
* Una jerarquía de clases no indica lo mismo que la jerarquía top-down
* Cuando se ejecuta el programa lo que tenemos son objetos que cooperan y que se crean dinámicamente durante la ejecución del programa
* Podemos pensar la interacción usuario/software de la misma manera
* Este mismo modelo nos permite entender (al menos en parte) otros modelos de computación: viendo a los objetos como proveedores de servicios, por ejemplo
* Este mismo modelo no asume objetos localizados en el mismo espacio de memoria (pueden estar distribuidos)

Impacto en como pensamos el software:

* Mientras que la estructura sintáctica es “lineal”, el programa en ejecución no lo es.

¿Qué es un objeto?

* Es una abstracción de una entidad del dominio del problema. Ejemplos: Persona, Producto, Cuenta Bancaria, Auto, Plan de Estudios….
* Puede representar también conceptos del espacio de la solución (estructuras de datos, tipos “básicos”, archivos, ventanas, iconos...)

Un objeto tiene:

* Identidad
  + Para distinguir un objeto de otro
* Conocimiento
  + En base a sus relaciones con otros objetos y su estado interno
* Comportamiento
  + Conjunto de mensajes que un objeto sabe responder

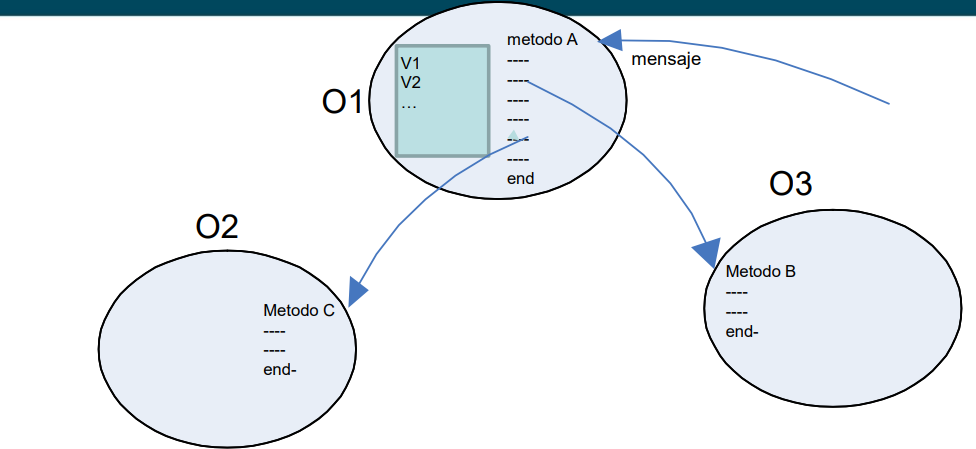
El estado interno:

* El estado interno de un objeto determina su conocimiento.
* El estado interno esta dado por:
  + Propiedades básicas (intrínsecas) del objeto. (dni, apellido, nombre)
  + Otros objetos con los cuales colabora para llevar a cabo sus responsabilidades.
* El estado interno se mantiene en las variables de instancia (v.i.) del objeto.
* Es privado del objeto. Ningún otro objeto puede accederlo. (¿Cuál es el impacto de esto?)

Variables de instancia:

En general las variables son REFERENCIAS (punteros) a otros objetos con los cuales el objeto colabora.

Algunas pueden ser atributos básicos



En el grafico O1 puede mandarle mensajes a O2 y O3 porque “los conoce”, o sea hay una variable en O1 que APUNTA a O2 y otra a O3 (o la misma variable que cambia de valor en diferentes momentos)

Comportamiento:

Un objeto se define en términos de su comportamiento.

El comportamiento indica qué sabe hacer el objeto. Cuáles son sus responsabilidades.

Se especifica a través del conjunto de mensajes que el objeto sabe responder: protocolo.



Implementación de comportamiento:

La realización de cada mensaje (es decir, la manera en que un objeto responde a un mensaje) se especifica a través de un método.

Cuando un objeto recibe un mensaje responde activando el método asociado.

El que envía el mensaje delega en el receptor la manera de resolverlo, que es privada del objeto



Envio de mensajes:

Para poder enviarle un mensaje a un objeto, hay que conocerlo.

Al enviarle un mensaje a un objeto, éste responde activando el método asociado a ese mensaje (siempre y cuando exista).

Como resultado del envío de un mensaje puede retornarse un objeto.

Lo que circula en los mensajes son objetos.

¿Cómo se especifica un mensaje?

* Nombre: correspondiente al protocolo del objeto receptor.
* Parámetros: información necesaria para resolver el mensaje.

Ejemplo:

cuenta.depositar(cantidad)

figura.dibujar()  
figuraGrande.rotar(45)

cuenta, figura, figuraGrande son variables que apuntan a un objeto que entiende el mensaje correspondiente

Métodos:

Que es un método?

* Es la contraparte funcional del mensaje.
* Expresa la forma de llevar a cabo la semántica propia de un mensaje particular (el cómo).

Un método puede realizar básicamente 3 cosas:

* Modificar el estado interno del objeto.
* Colaborar con otros objetos (enviándoles mensajes).
* Retornar y terminar.

Nunca hay script entrada y salida.

Programamos clases, no objetos.

Las clases son moldes

Hay librerías de clases y hay librerías de funciones

Como creamos objetos?

Comúnmente se utiliza la palabra new para instanciar nuevos objetos.

Los objetos se van creando bajo necesidad.

Ej: voy al banco y quiero abrir una cuenta, se crea un objeto cuenta, y quiero abrir una caja de ahorro, se crea el objeto de la caja de ahorro. Se conecta todo como un grafo que el banco conoce la cuenta, la cuenta la caja de ahorro, el banco conoce la caja de ahorro, etc.