

Es Python el lenguaje más usado por los alumnos de la asignatura?

Python tiene mucho de Orientación a Objetos, pero... es “error-prone”...

En realidad se usa habitualmente “sin saber” de Orientación a Objetos porque sólo se utiliza como “aglutinador” de librerías (que normalmente se han desarrollado en C/C++ (Python es demasiado lento))

## Qué es la programación Orientada a Objetos?

Un modo de desarrollar software en el que

Nos focalizamos en escribir “piezas” razonablemente reducidas de modo que podemos “controlarlas” con facilidad (no sobrepasan nuestra capacidad mental para tener en cuenta detalles o estructuras)

Estas piezas se plantean de un modo prácticamente independiente del objetivo final -la aplicación informática-, sino más bien pensando en su propia coherencia (en este sentido representarán objetos reales o conceptos claramente definidos). Obviamente esto no es “del todo” cierto, pero sí lo es en buena medida (pensemos en “Persona” para un hospital o para una universidad).

Los objetos “se apoyan” en otros objetos para todo aquello que resulta complejo dentro de su planteamiento (pensemos en “Persona” en el ámbito universitario, y su “ExpedienteAcadémico”).

La aplicación informática, como otro objeto cualquiera, será una composición también razonablemente sencilla en el que las complejidades se delegan en otros objetos.

## Cómo funciona un programa Orientado a Objetos?

En definitiva un programa orientado a objetos es una colección de objetos colaborando entre ellos con un determinado fin o proporcionando unas determinadas capacidades.

Los objetos se comunican entre sí “enviándose mensajes” (“pidiendo servicios”, “dando ordenes”... como queramos verlo.

La terminología “enviar mensajes” puede concretarse de diferentes maneras: desde un modo estricto, es decir, enviar mensajes con estructura predeterminada a través de un canal de comunicaciones, hasta ejecutar llamadas a subrutinas o compartir datos en un área de memoria común.

En todo caso se trata de flujos de información que estarán determinados/definidos por sus “interfaces” (la parte que los objetos muestran al exterior).

## Cómo se escribe un programa Orientado a Objetos?

Escribiendo clases de las que se instancian Objetos.

En la terminología original se hablaba de Tipos y Objetos y hoy hablamos de Clases y Objetos. Desde un punto de vista “mecánico”, los Objetos son realizaciones concretas (instancias) de “Tipos complejos” del mismo modo que las variables son realizaciones concretas de tipos simples (en terminología Java, “tipos primitivos”).

Para programar con Objetos basta con hacerlo en un entorno que los proporcione (por ejemplo con JavaScript dentro de un navegador web). Esto es lo que se denomina programación “basada” en objetos (actualmente ampliada como programación “basada en prototipos”). Si además tenemos la capacidad de definir nuevos “tipos complejos” (nuevas Clases), hablamos de programación Orientada a Objetos.

## Porqué surge la programación Orientada a Objetos?

Esta idea de “tipo complejo” ya venía en cierto modo practicándose desde antes de definirse la Orientación a Objetos. Un punto concreto en donde podía apreciarse bien era en el planteamiento de lo que se conoce como “Tipos Abstractos”. Los Tipos Abstractos son estructuras de datos que se definen por las operaciones que se pueden hacer sobre ellos independientemente de cómo se concreten en la memoria del ordenador (p.ej. una “pila” o “LIFO” con las operaciones “push” y “pop” .

Los “buenos programadores” estructuraban su código de un modo similar a lo que hoy es la definición de clases, y la observación de que su formalización daba lugar a nuevos conceptos y capacidades de gran potencia estructural, llevó a considerarse un nuevo paradigma.

## Cómo se estructura un programa Orientado a Objetos?

La Orientación a Objetos potenció enormemente la arquitectura de aplicaciones. Independientemente de estructuraciones más específicas, un concepto básico es el de la estructura en capas (tiers), que en su versión básica consiste en tres capas, lo que se conoce como modelo MVC (o ECB)

Las capas “exteriores” pueden desdoblarse para dar lugar a un modelo de hasta 5 capas, donde ahora las más externas son aplicaciones ligeramente acopladas al núcleo de 3 capas.

La parte M de esta estructura en capas, enlaza directamente con lo que se trata en este curso, y además nos permite introducir un “cierto” paralelismo entre las clases M y las tablas de una base de datos.