

Curso de Desarrollo en Lenguaje Python para Inteligencia Artificial (Málaga)

M.374.001.001

30 de marzo 2021 09:30-13:30

Modulo 2 – Tema 3

Spiros Michalakopoulos



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

 polo de contenidos
digitales



**Ayuntamiento
de Málaga**

Módulo 2: Programación orientada a objetos (POO) en Python

Desarrollo en Lenguaje Python para Inteligencia Artificial

Año de realización: 2021

PROFESOR

Spiros Michalakopoulos

spiros.eoi@gmail.com

<https://www.linkedin.com/in/spiros-michalakopoulos>



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

POO en Python

Índice

1. Bases de la programación orientada a objetos
2. Clases, objetos, atributos, instancias, propiedades
3. Métodos, métodos de las colecciones
4. Encapsulación, propiedades
5. Herencia, abstracción, polimorfismo
6. Errores y excepciones
7. Paquetes y distribución de software





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

Tema 3

Herencia, abstracción, polimorfismo.



Herencia (1/2)

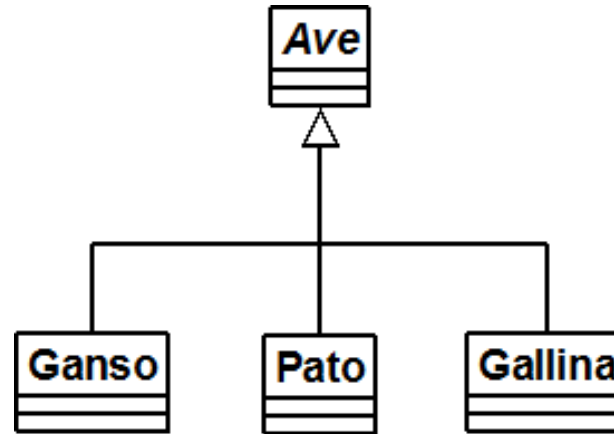
Reutilización de código

La POO tiene como uno de sus objetivos principales la **reutilización** de código. La **herencia** (o **generalización**, o **especialización**) es un ejemplo de ello. Consiste en la definición de una clase utilizando como base una clase ya existente. La nueva clase derivada tendrá todas la características de la clase base y ampliará el concepto de ésta, es decir, tendrá todos los atributos y métodos de la clase base.



Herencia (2/2)

“es un” (“is-a”)



En POO la **herencia** significa que entre las dos clases existe una relación del tipo "**es un**", ("**is-a**").

La clase cuyas características se heredan se conoce como **superclase** (o una **clase base** o una **clase padre**). La clase que hereda se conoce como **subclase** (o una **clase derivada**, **clase extendida** o **clase hija**).



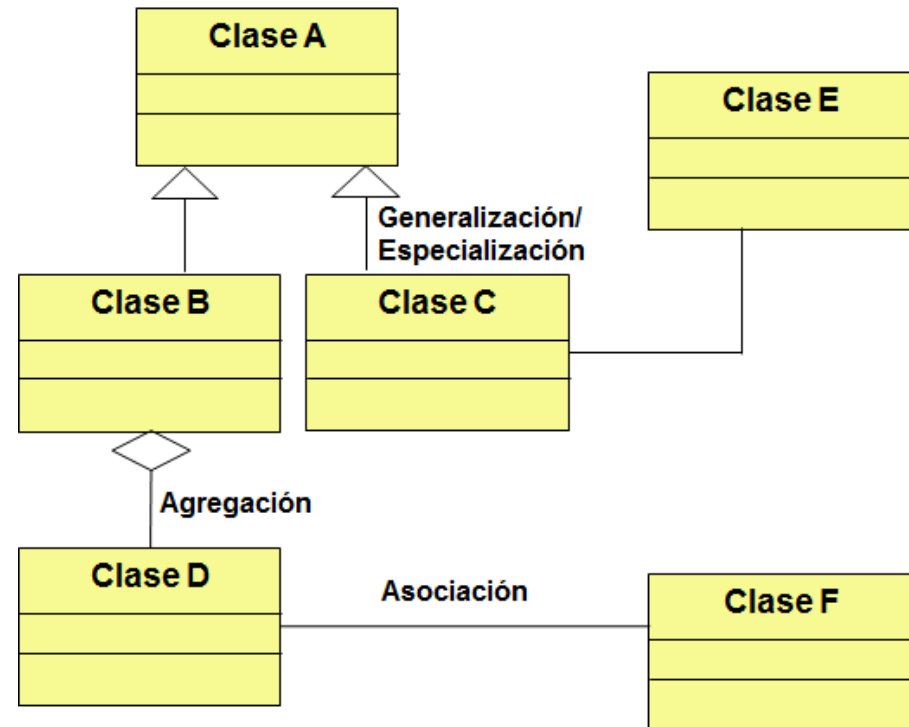
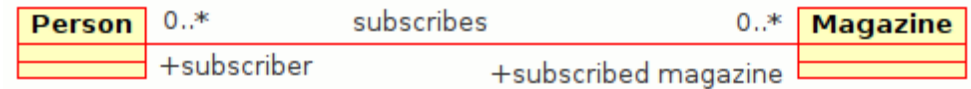
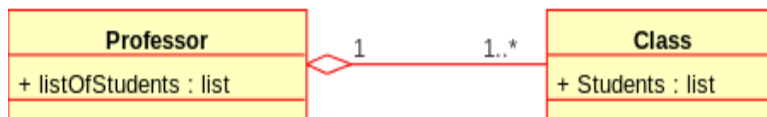
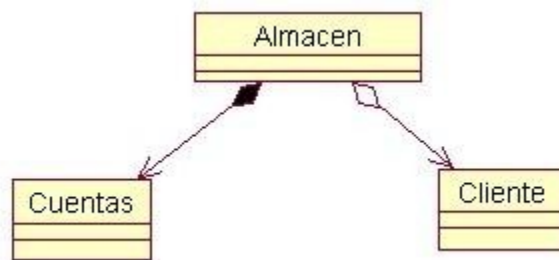
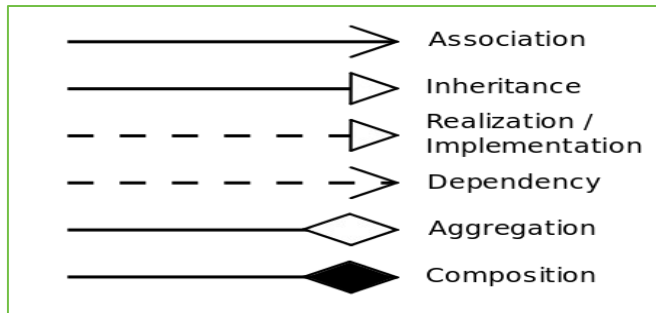
Diagramas de clases

Diagramas estáticos de clases y sus relaciones

Un **diagrama de clases** es un diagrama de estructura estático que muestra las diferentes **clases** de objetos que componen un sistema y **cómo se relacionan** unas con otras. Las relaciones estáticas más utilizadas son las de **asociación**, **composición** y **herencia**. Los componentes principales de un diagrama de clases son las clases y todas las posibles relaciones existentes entre éstas (**asociación**, **herencia** (**generalización-especialización**), y **agregación/composición**).



Diagramas de clases





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

POO en Python

Interfaces y clases abstractas (1/3)

¿Qué es una interfaz?

Una **interfaz** (**Interface** en inglés) declara un conjunto de **métodos** y **constantes** sin especificar su implementación para ningún método. Cuando una clase exige la implementación de una interfaz, debe proporcionar la implementación para **todos los métodos** declarados en la interfaz.





Interfaces y clases abstractas (2/3)

¿Qué es una clase abstracta?

En una **clase abstracta** (**ABC**: **a**bstract **b**ase **c**lass) es posible definir **métodos abstractos**, los cuales se caracterizan por el hecho de que no pueden ser implementados en la clase base. De ellos, solo se escribe su **signatura** en la superclase, y su funcionalidad – **polimórfica**– tiene que indicarse en las clases derivadas (subclases).



Interfaces y clases abstractas (3/3)

Interfaces vs. clases abstractas

1. Una **clase abstracta** puede heredar o extender **cualquier** clase, mientras que una **interfaz** solamente puede extender o implementar **otras interfaces**.
2. Una **clase abstracta** puede heredar de **una sola clase** mientras que una **interfaz** puede extender **varias interfaces** de una misma vez.
3. Una **clase abstracta** puede tener métodos que sean abstractos **o que no lo sean**, mientras que las **interfaces sólo** y exclusivamente pueden definir métodos abstractos.