

A continuació trobaràs enllaços a recursos i tutorials, que t'ajudaran a resoldre l'activitat d'aquest apartat:

Tasca -> [POO 2](#)



L'orientació a objectes és un paradigma de la programació que té uns efectes i una profunditat d'ús tan immensos com la mateixa imaginació i capacitat d'observació de l'ésser humà.

En aquest tema analitzarem conceptes una mica més avançats sobre aquesta qüestió.

Abstracció

De vegades hem de representar conceptes abstractes en programació, i no només pel fet de representar-los, sinó pel fet encara més rellevant que separar l'essència dels detalls concrets ens permetrà realitzar programes més testeables, extensibles, comprensibles i modulars.

Classe Abstracta

-> [abstract class](#)

Abstracció a Java

-> [What is Abstraction in OOPs? Java Abstract Class & Method](#)

Abstracció en PHP

-> [Clases abstractas en PHP 8](#)

Interfícies

Ja hauràs vist força referència a interfícies (interface) en alguns dels articles anteriors, però, què són i en què es diferencien de les classes abstractes?

-> [Interfaces](#)

Interfícies a Java

-> [What is an Interface?](#)

Interfícies a PHP

-> [Interfaces en PHP](#)

¿Quina diferència hi ha entre classes abstractes i interfícies?

Compte, que algunes diferències subtils depenen del llenguatge de programació en si però, en general...

->[PHP Interface vs Abstract class](#)

Standards de codificació

Com bé hauràs notat, cada cop els programes que fas seran més grans. Per aquest motiu, és important posar-se d'acord en detalls com maneres d'anomenar les coses i organitzar codi. Per això existeixen estàndards com el que et proposa PHP-FIG.

Standards de codificació a PHP

->[PSR-1: Basic Coding Standard](#)

✉ [Contacteu amb el suport del lloc](#)

Heu iniciat sessió com a [Mario Gomez \(Surt\)](#)

[Política de Cookies](#) - [Política de Privacitat](#) - [Condicions Generals d'Us](#)

Funciona amb [Moodle](#)



Unió Europea
Fons Europeu
de Desenvolupament Regional

[Privacy Policy](#)