

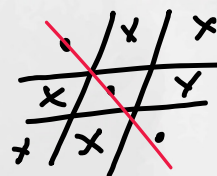
OOP in PHP

LA PROGRAMMAZIONE ORIENTATA AGLI OGGETTI (OOP) MODELLA LA REALTÀ TRAMITE LE "CLASSI", CHE A LORO VOLTA DEFINISCONO:

- **PROPRIETÀ** (=CARATTERISTICHE DI UN OGGETTO)
- **METODI** (=COMPORTAMENTI, AZIONI CHE L'OGGETTO SA E PUÒ FARE ESEGUIRE)

Classe

GLI OGGETTI SONO CREATI A PARTIRE DALLE CLASSI TRAMITE L'INSTANZIAZIONE



Oggetto

È COMPOSTO DA
PROPRIETÀ (DATI)
METODI (FUNZIONI)

È UNA **COPIA PERSONALIZZABILE** DELLA CLASSE DA CUI HA ORIGINE

PUÒ ESSERE INIZIALIZZATO (PERSONALIZZATO) USANDO LA KEYWORD "**__CONSTRUCT**"

È DOTATO DI UNA **ISTANZA UNICA**, PUNTABILE (IDENTIFICABILE) TRAMITE LA PSEUDO-VARIABILE "**\$THIS**"

CREDITI

Contenuti
PASQUALE ALFANO

[linkedin.com/in/pasqualealfano/](https://www.linkedin.com/in/pasqualealfano/)

@ibisredibis

github.com/halfpass25/

Grafica

LORENZO ALFANO

[linkedin.com/in/lorenzcoalfano/](https://www.linkedin.com/in/lorenzcoalfano/)

[behance.net/LorenzoAlfano](https://www.behance.net/LorenzoAlfano)

Incapsulamento

NASCONDE I DETTAGLI INTERNI DELLA CLASSE MA **ESPONE** PROPRIETÀ (CARATTERISTICHE, DETTE ANCHE "DATI") E METODI (=FUNZIONI) TRAMITE INTERFACCIA PUBBLICA, ED IMPEDISCE MODIFICHE INVOLONTARIE DEI VALORI ASSEGNATI ALLE PROPRIETÀ.

VISIBILITÀ (PER PROPRIETÀ O METODI)

Public

ACCESSIBILE **GLOBALMENTE** PER LETTURA/SCRITTURA

Protected

ACCESSIBILE **SOLO** DALLE CLASSI **EREDITANTI**

Private

ACCESSIBILE **SOLO** ALLA CLASSE IN CUI È **DICHIARATO** TALE (TIPICAMENTE, UNA CLASSE GENITRICE)

Ereditarietà

UNA CLASSE PUÒ "EREDITARE" PROPRIETÀ E METODI DA UN'ALTRA CLASSE DETTA "GENITRICE". IL MECCANISMO CONSENTE DI AVERE CODICE RIUSABILE E BEN ORGANIZZATO. UNA CLASSE, OVVIAMENTE, OLTRE AD EREDITARE PUÒ DEFINIRE I SUOI PROPRI METODI E PROPRIETÀ.

MA PHP OFFRE SOLO **EREDITARIETÀ SINGOLA**: COME SUPERARE TALE LIMITE?

PER SUPERARE I LIMITI DELLA EREDITARIETÀ SINGOLA, POTETE USARE LE SEGUENTI TECNICHE, **SINGOLARMENTE O COMBINANDOLE INSIEME**

Poliformismo

PERMETTE AGLI "OGGETTI" DI CAMBIARE FORMA SEMPLICEMENTE ESTENDENDO O SCAVALCANDO METODI GIÀ ESISTENTI. AD ESEMPIO: LE CLASSI INDIAN E BRITISH POSSONO ESTENDERE (EREDITANO DA) LA CLASSE SPEAKER MA CIASCUNA AVRÀ POI IL SUO PROPRIO SPOKEN_LANGUAGE.

Astrazione

L'ASTRAZIONE MIRA A RIDURRE LA COMPLESSITÀ, MOSTRANDO SOLO CIÒ CHE SERVE IN UN DATO CONTEST. AD ESEMPIO, UN UTENTE NON HA BISOGNO DI SAPERE COME FUNZIONA INTERNAMENTE UNO SPORTELLLO BANCOMAT, GLI BASTA SAPERE COME PUÒ INTERAGIRE CON ESSO.

Ereditarietà verticale

(KEYWORD: **EXTENDS**) È L'EREDITARIETÀ BASE OFFERTA DA PHP; PER INDICARE AD UNA "CLASSE FIGLIA" CHE EREDITERÀ DA UNA "CLASSE GENITRICE", UTILIZZATE LA KEYWORD "EXTENDS".

Ereditarietà orizzontale

NOTA ANCHE COME "TRAITS" (KEYWORD: **USE**) TRAMITE I TRAITS, UNA CLASSE FIGLIA PUÒ RIUTILIZZARE CODICE PROVENIENTE DA PIÙ DI UNA CLASSE SENZA RICHIEDERE CHE QUESTE ULTIME SIANO FIGLIE DI UNA CLASSE GENITRICE. PER UTILIZZARE I TRAITS:

1 - DEFINITE UN TRAIT (TIPICAMENTE, È UNA COLLEZIONE DI METODI)

2 - PER INCLUDERLO IN UNA CLASSE, UTILIZZATE LA KEYWORD "USE"

3 - I METODI ESPOSTI DAI TRAITS DIVENGONO DISPONIBILI ALLA CLASSE CHE LI USA, COME SE FOSSE DEFINITI NELLA CLASSE STESSA

Interfacce

UNA "INTERFACCIA", IN PHP, SPECIFICA UN INSIEME DI METODI CHE DEVONO ESSERE IMPLEMENTATI NELLE CLASSI CHE FANNO USO DELL'INTERFACCIA; IN PRATICA, L'INTERFACCIA DEFINISCE UN CONTRATTO CHE DEVE ESSERE RISPETTATO. L'INTERFACCIA DEFINISCE SOLO IL NOME DEI METODI, I QUALI, AVRANNO UNA SPECIFICA IMPLEMENTAZIONE ALL'INTERNO DELLE CLASSI CHE ADERISCONO AL CONTRATTO. UN ESEMPIO BANALE MA CHIARISSIMO: SE NELL'INTERFACCIA AVETE DICHIARATO UN METODO DI NOME "CALCOLAAREA", POTRETE POI AVERE SINGOLE CLASSI CHE CALCOLANO L'AREA DI UN CERCHIO O DI UN QUADRATO ED OGNUNA DI QUELLE CLASSI, IMPLEMENTERÀ IL METODO "CALCOLAAREA" TENENDO CONTO DELLA SPECIFICA FIGURA GEOMETRICA SU CUI LA CLASSE OPERA. IL NOME DEL METODO RIMANE DUNQUE SEMPRE LO STESSO, CAMBIA SOLO LA SUA IMPLEMENTAZIONE DI CLASSE IN CLASSE.