

OOP

Object Oriented Programming

Connessione a DBMS

PDO

OOP – Object Oriented Programming

- **Classe**: modello generico (template) che rappresenta (incapsula) le caratteristiche comuni di una entità
- **Oggetto (o istanza)**: rappresentazione concreta e specifica di un elemento che appartiene alla classe
- **Proprietà**: o attributi (variabili) della classe. Il valore determina le caratteristiche o lo stato dei singoli oggetti istanziati da quella classe
- **Metodo**: funzionalità (funzioni) della classe.
- **Visibilità** di metodi e proprietà
 - **public**: accessibile da qualunque posto (default)
 - **protected**: accessibile soltanto dalla classe stessa e dalle classi che la ereditano
 - **private**: accessibile solo dalla classe in cui è definito

Classi e oggetti

- Costruttore: il metodo `__construct`
- Si può specificare un metodo distruttore `__destruct`
- `$this` per accedere alle variabili e ai metodi della classe nell'istanza
- Creazione di un nuovo oggetto: `new`
- Notazione `->` per accedere alle proprietà e ai metodi di un oggetto
(es: `$o->proprietà`, `$o->metodo()`)
- Notazione `::` per accedere a metodi statici

Attenzione: quando si accede ad una proprietà **NON** si usa `$` prima del nome della proprietà

- Video utili:
- 1. <https://www.youtube.com/watch?v=Dm1SNnxQxHE&t=75s>
- 2. <https://www.youtube.com/watch?v=IIZu0vztMlo&t=31s>

Esempio base

classe_base.php

```
<?php
//definizione di classe
class Esempio {
    function saluto(){
        echo "Ciao";
    }
}
//istanza di un nuovo oggetto della classe Esempio
$e = new Esempio();
$e->saluto();
?>
```

Esempio completo

classe.php

```
<?php

class Persona {
    // proprietà
    private $nome = "";

    //costruttore (metodo magico)
    public function __construct($nome){

        //inizializzo la proprietà name
        $this -> nome = $nome;
    }

    //metodo
    public function getName() {
        //$this rappresenta l'oggetto che sarà costruito a runtime
        return $this -> nome;
    }
}

?>
```

```
<?php

//istanzio 2 nuovi oggetti della classe Persona

$studente = new persona("Mario Rossi");

$docente = new persona("Giacome Verdi");

?>

Studente:
    <?php echo $studente->getName()?><br>

Docente:
    <?php echo $docente->getName()?><br>
```

OOP

Object Oriented Programming

Connessione a DBMS

PDO

Accesso al Database

- Data-access abstraction layer: livello di astrazione per l'accesso ai dati
- PDO (PHP Data Objects), basata sulla programmazione ad oggetti, introdotta dalla versione 5 di PHP.
- Una classe e un insieme di sotto-classi derivate che permette di creare applicazioni portabili indipendenti dal DBMS.
- Se non abilitato, nel php.ini, togliere il commento alla riga (e a tutte quelle relative al db MySQL):
- `extension=php_pdo.dll`

Esempio completo di connessione

PDO_esempio_completo.php

```
<?php
$host = "localhost";      $user = "root";      $pass = "";      $dbname = "ifts";
$dsn = 'mysql:host=' . $host . ';dbname=' . $dbname;

try {
    $connessione = new PDO($dsn, $user, $pass);
    $st = $connessione->prepare("SELECT * FROM regioni WHERE ID_regione = :id ");
    $id = 2;
    $st->bindParam(':id',$id,PDO::PARAM_INT);
    $st->execute();
    $records=$st->fetchAll(PDO::FETCH_ASSOC);

} catch(PDOException $e) {
    die("Errore durante la connessione al database!: ". $e->getMessage());
}

echo "estratti n. record = ".count($records);
?>
```