Classe "Concerto" in PHP: Struttura, Metodi, e Gestione degli Attributi

**Introduzione:**

La classe "Concerto" in PHP è progettata per gestire informazioni relative ai concerti, fornendo un'interfaccia per creare, modificare, visualizzare e cancellare record di concerti all'interno di una tabella di database. Questa classe è strutturata in modo da garantire l'incapsulamento dei dati attraverso l'uso di attributi privati, con i metodi Get e Set per accedere o modificare tali attributi.

**Attributi della Classe Concerto**

La classe "Concerto" ha i seguenti attributi:

Id: intero; variabile auto-incrementale, non modificabile, chiave primaria.

Codice: intero; codice appartenente al concerto.

Titolo: stringa; titolo del concerto.

Descrizione: stringa; descrizione del concerto.

Data: date time; data del concerto.

**Costruttore:**

Il costruttore come parametri di Input riceverà: codice, titolo, descrizione e data ma non l’id, dato che si tratta di un valore auto-incrementale e non modificabile che verrà creato automaticamente.

**Metodi di istanza:**

Precisazione: in tutti i metodi Set\_\* viene opportunamente controllata la correttezza del dato inserito, nel caso in cui il dato sia nella forma errata viene chiesto di reinserire il valore.

* ***Delete()***: Questo metodo elimina il record corrispondente all'oggetto "Concerto" dalla tabella.

Parametri di Input: nessun parametro.

Valore di Ritorno: nessun valore di ritorno.

* ***Update()***: Questo metodo permette di aggiornare i dati del record corrispondente all'oggetto "Concerto" nella tabella.

Parametri di Input: nuovi valori per il codice, il titolo, la descrizione e la data del concerto.

Valore di ritorno: nessun valore di ritorno.

* ***Save()***: E’ un metodo privato, ueviene chiamato internamente dalla classe quando si desidera salvare o aggiornare i dati del concerto.

Parametri di Input: i parametri della classe “Concerto” ad esclusione dell’id.

Valore di ritorno: nessun valore di ritorno.

* ***Get\_Id()***: metodo pubblico per far ritornare l’id del concerto.

Parametri di Input: nessun parametro.

Valore di Ritorno: un intero che rappresenta l’id del concerto.

* ***Set\_Codice()***: metodo pubblico per impostare il codice del concerto.

Parametri di input: un intero che rappresenta il codice del concerto.

Valore di ritorno: nessun valore di ritorno.

* ***Get\_Codice()***: metodo pubblico per far ritornare il codice del concerto.

Parametri di Input: nessun parametro.

Valore di Ritorno: un intero che rappresenta il codice del concerto.

* ***Set\_Titolo()***: metodo pubblico per impostare il titolo del concerto.

Parametri di input: una stringa che rappresenta il titolo del concerto.

Valore di ritorno: nessun valore di ritorno.

* ***Get\_Titolo()***: metodo pubblico per far ritornare il titolo del concerto.

Parametri di Input: nessun parametro.

Valore di ritorno: una stringa che rappresenta il titolo del concerto.

* ***Set\_Descrizione()***: metodo pubblico per impostare la descrizione del concerto.

Parametri di Input: una stringa che rappresenta la descrizione del concerto.

Valore di ritorno: nessun valore di ritorno.

* ***Get\_Descrizione():*** metodo pubblico per far ritornare la descrizione del concerto.

Parametri di Input: nessun parametro.

Valore di ritorno: una stringa che rappresenta la descrizione del concerto.

* ***Set\_Data()***: metodo pubblico per impostare la data del concerto.

Parametri di Input: un date time che rappresenta la data del concerto.

Valore di ritorno: nessun valore di ritorno.

* ***Get\_Data():*** metodo pubblico per far ritornare la data del concerto.

Parametri di Input: nessun parametro.

Valore di ritorno: la data del concerto.

**Metodi statici:**

* **Create()**: Questo metodo crea un nuovo record nella tabella dei concerti e restituisce un oggetto "Concerto" corrispondente al nuovo record o un identificatore (id) del nuovo record.

Parametri di Input: I parametri della classe “Concerto” ad esclusione dell’id.

Valore di Ritorno: ritorna un oggetto di tipo "Concerto" se la creazione è andata a buon fine altrimenti solleva un'eccezione e non crea l’oggetto.

* **Find($id):** Questo metodo trova e restituisce un oggetto "Concerto" corrispondente all'identificativo (id) specificato.

Parametri di Input: L'identificatore (id) del concerto da cercare.

Valore di Ritorno: Un oggetto "Concerto" corrispondente all'identificativo specificato o NULL se il concerto non è stato trovato.

* ***Show()***: Questo metodo restituisce un oggetto “Concerto”.

Parametri di Input: id record.

Valore di Ritorno: un oggetto “Concerto”.

link:

<https://www.php.net/manual/en/pdo.connections.php>

Creazione classe per creare la connessione con il database

Metodi php:

<https://www.php.net/manual/en/class.pdo.php>

fetch all(prende tutti i record):

public [fetchAll](https://www.php.net/manual/en/pdostatement.fetchall.php)(int $mode = PDO::FETCH\_DEFAULT): array

fetch object(ritorna una colonna come un oggetto):

public [fetchObject](https://www.php.net/manual/en/pdostatement.fetchobject.php)(?string $class = "stdClass", array $constructorArgs = []): object|false

Metodo find(prima bisogna connettersi):

public function findById(int $userId)

{

$pdos = $this->dBase->prepare('SELECT user\_id AS objectId, name, description, password, active, created,

last\_update AS lastUpdate FROM user WHERE user\_id = :id');

$pdos->bindParam(':id', $userId, \PDO::PARAM\_INT);

$pdos->execute();

return $pdos->fetchObject('\DomainObjects\User', array($this->password));

}

Fare il metodo create per lunedì.

connessione