

OOP in PHP ตอนที่ 3

สวัสดีครับ สำหรับในตอนี่ 3 นี้เราจะพูดถึงการสืบทอดและการประยุกต์ใช้กับฐานข้อมูล MySQL กัน โดยตัวจาก 2 บทที่แล้วนั้นเรามาทบทวนคำเฉพาะกับสักเล็กน้อยว่า เรามีความเฉพาะต่าง ๆ ดังนี้

- แม่แบบ (Class)
- ลูกแบบ (Sub-Class)
- วัตถุ (Object)
- การส่งข้อความ (Messaging)
- ข้อความ (Message)
- การห่อหุ้มข้อมูล(Encapsulation)
- การสืบทอด (Inheritance)
- การทำหลายรูปแบบ (Polymorphism)
- คุณสมบัติ (Properties)
- แบบแผนพฤติกรรม (Method Behavior)
- พฤติกรรมพื้นฐาน (Constructor Method Behavior)

โดยจากตอนที่แล้วนั้นก็เขียนโปรแกรมแบบที่เกี่ยวกับวันและเวลา โดยนำฟังก์ชันพื้นฐานทั่วไป นำมาใส่ในตัวแม่แบบแล้วนำไปใช้งานโดยการนำมาใช้นั้นจะช่วยให้การทำความเข้าใจตัวโปรแกรมง่ายขึ้น

การสืบทอด (Inheritance)

การสืบทอดในการเขียนโปรแกรมเชิงวัตถุ นั้น เป็นการทำให้แม่แบบแต่ละแบบนั้นมีความเฉพาะเจาะจงลงไปมากขึ้นตามความละเอียดและจำเพาะของคุณสมบัติและแบบพฤติกรรม ตัวอย่างเช่นแม่แบบมนุษย์ มีการสืบทอดแม่แบบมาเป็นเพศชาย และเพศหญิง นั้นเอง โดยการเขียนตัวชุดคำสั่งนั้นเราจะใช้คำเฉพาะว่า **extends** แล้วตามด้วยชื่อของแม่แบบที่ตัวเองสืบทอดมา ดังตัวอย่างต่อไปนี้

```
<?php
class human {
    function Talk() {
        echo "Hello World";
    }
}
class male extends human {
    . . . .
}
class female extends human {
    . . . .
}
?>
```

โดยการสืบทอดใน PHP ในรุ่น 4 นั้นจะนำคุณสมบัติและแบบพฤติกรรมทั้งหมดของแม่แบบที่ตนเองสืบทอดมาทั้งหมด โดยการเรียกใช้นั้นเหมือนกับว่าคุณสมบัติและแบบพฤติกรรมนั้นถูกสร้างและทำงานอยู่บนแม่แบบที่ตัวเองมีทั้งหมด ทำให้การสืบทอดมานั้นมีประโยชน์ในการต่อเติมเสริมความสามารถที่ขาดหายไปของแม่แบบที่ตัวลูกแบบนั้นต้องการเพิ่มเติมและทำให้คุณสมบัติต่างๆ ของลูกแบบนั้นดูมีความเฉพาะมากขึ้นด้วย และลดความซ้ำซ้อนของการใช้งานคุณสมบัติและพฤติกรรมหลายๆ อย่างที่เหมือน ๆ กันได้ด้วย

การประยุกต์ใช้กับงานด้านติดต่อฐานข้อมูล

ดังที่ได้กล่าวไปแล้วว่าในตอนนี้อเราจะเขียนตัวแม่แบบเพื่อนำไปสร้างวัตถุสำหรับติดต่อกับฐานข้อมูล เรามาดูกันเลยว่าเราต้องเตรียมคุณสมบัติและแบบพฤติกรรมอะไรบ้าง

แม่แบบ **mysql_config** เป็นแม่แบบที่ใช้ในการเก็บค่าที่จำเป็นสำหรับการเชื่อมต่อกับฐานข้อมูล เพื่อเอาวัตถุที่ได้จากแม่แบบนี้โยนไปให้กับแม่แบบ **mysql_operator** เพื่อทำงานกับฐานข้อมูลต่อไป โดยมีคุณสมบัติอยู่ 1 อย่างคือ

- **dbconfig** ที่เป็นข้อมูลชนิด array ที่เก็บค่า config ต่าง ๆ ทั้ง 6 รายการซึ่งได้แก่ hostname, database, username, password, character set และ collation connection อยู่ภายใน

และมีพฤติกรรมต่าง ๆ อีก 12 พฤติกรรมดังนี้

- **mysql_config** เมื่อเริ่มสร้างวัตถุให้ทำการนับเวลาการ Query ข้อมูลและจำนวนการ Query
- **get_configdb** ส่งข้อมูลการตั้งค่าของฐานข้อมูลออกมา
- **get_querydb_counter** เรียกดูจำนวนชุดคำสั่ง SQL ที่ได้ทำงาน
- **get_querydb_timer** เรียกดูเวลาในการดำเนินการชุดคำสั่ง SQL
- **set_character_set** ตั้งค่า Character Set
- **set_collation_connection** ตั้งค่า Collation Connection
- **set_database** ตั้งค่า Database ที่ใช้
- **set_hostname** ตั้งค่า Hostname ที่ใช้
- **set_username** ตั้งค่า Username
- **set_password** ตั้งค่า Password
- **set_querydb_counter** ตั้งค่าตัวนับจำนวนชุดคำสั่ง SQL ที่ได้ทำงาน
- **set_querydb_timer** ตั้งค่าตัวนับเวลาในการดำเนินการชุดคำสั่ง SQL

แม่แบบ **mysql_operator** เป็นแม่แบบที่ใช้ในการทำงานกับฐานข้อมูล MySQL โดยที่มีการสืบทอด จากแม่แบบ **mysql_config** ลงมา โดยมีคุณสมบัติอยู่ 4 อย่างคือ

- **link** เก็บสถานะการเชื่อมต่อกับฐานข้อมูล
- **result** เก็บผลการทำงานของการเรียกใช้หรือจัดการข้อมูลในฐานข้อมูล
- **mysql_version** เก็บค่ารุ่นของฐานข้อมูล MySQL
- **string_error** เก็บข้อความแจ้งข้อผิดพลาดของการทำงานของคำสั่ง SQL

และมีพฤติกรรมต่าง ๆ อีก 27 พฤติกรรมดังนี้

- **mysql_operator** เมื่อเริ่มต้นสร้างวัตถุ จะรับค่าต่าง ๆ เพื่อตั้งค่าสำหรับติดต่อกับฐานข้อมูลผ่านวัตถุที่ได้จากแม่แบบ **mysql_config**

- `opendb` พุทธิตกรรมสำหรับเปิดการเชื่อมต่อฐานข้อมูล MySQL
- `closed` พุทธิตกรรมสำหรับปิดการเชื่อมต่อฐานข้อมูล MySQL
- `querydb` พุทธิตกรรมสำหรับเรียกใช้หรือสั่งให้ฐานข้อมูลทำงานตามรูปแบบคำสั่งภาษา
- `get_error_msg` พุทธิตกรรมสำหรับส่งข้อผิดพลาดของการทำงานออกมา
- `get_last_insert_id` พุทธิตกรรมสำหรับส่งคืนค่าหมายเลข `AUTO_INCREMENT` ที่สร้างจากการ `INSERT` ของ SQL
- `get_mysql_version` พุทธิตกรรมสำหรับสำหรับเรียกค่ารุ่นของฐานข้อมูล MySQL
- `get_resultdb` พุทธิตกรรมสำหรับส่งค่าผลการทำงานของการเรียกใช้ฐานข้อมูล
- `get_resultdb_atrow` พุทธิตกรรมสำหรับส่งค่าผลการทำงานของการเรียกใช้ฐานข้อมูลมาทีละแถวตามที่กำหนดไว้
- `get_resultdb_fetch_data` พุทธิตกรรมสำหรับส่งค่าผลการทำงานของการเรียกใช้ฐานข้อมูลโดยทำการแยกข้อมูลออกมาเป็น Array
- `get_resultdb_num_fields` พุทธิตกรรมสำหรับนับจำนวนฟิลด์ข้อมูลที่ส่งมาในการเรียกใช้ฐานข้อมูลในแต่ละครั้ง
- `get_resultdb_num_rows` พุทธิตกรรมสำหรับนับจำนวนแถวข้อมูลที่ส่งมาในการเรียกใช้ฐานข้อมูลในแต่ละครั้ง
- `has_data_in_table` พุทธิตกรรมสำหรับตรวจสอบข้อมูลในตารางว่ามีอยู่หรือไม่
- `list_fields` พุทธิตกรรมสำหรับเรียกรายการฟิลด์ในตารางออกมา
- `num_rows_in_table` พุทธิตกรรมสำหรับนับจำนวนแถวข้อมูลทั้งหมดในตาราง
- `querydb_counter_decrement` พุทธิตกรรมสำหรับลดค่าการนับจำนวนคำสั่งที่เข้าไปการเรียกฐานข้อมูล
- `querydb_counter_increment` พุทธิตกรรมสำหรับเพิ่มค่าการนับจำนวนคำสั่งที่เข้าไปการเรียกฐานข้อมูล
- `querydb_transac_begin` พุทธิตกรรมสำหรับเริ่มต้นการทำ transaction ในฐานข้อมูล
- `querydb_transac_check_for_rollback_or_commit` พุทธิตกรรมที่ใช้ในการทดสอบการทำงานของชุดคำสั่งหรือกลุ่ม transaction ต่าง ๆ ว่ามีข้อผิดพลาดในการทำงานหรือไม่ ถ้ามีจะทำการ rollback ข้อมูลกลับสู่สภาพเดิม แต่ถ้าไม่มีข้อผิดพลาดจำทำการ commit ข้อมูลลงฐานข้อมูล
- `querydb_transac_commit` พุทธิตกรรมสำหรับการทำ commit ข้อมูลลงฐานข้อมูล
- `querydb_transac_rollback` พุทธิตกรรมสำหรับการทำ rollback ข้อมูลในฐานข้อมูล
- `querydb_transac_set_autocommit_to_zero` พุทธิตกรรมสำหรับการตั้งค่า autocommit ไม่ให้ใช้งาน
- `reset_querydb` พุทธิตกรรมสำหรับรับข้อผิดพลาดของการทำงาน
- `set_configdb` พุทธิตกรรมสำหรับตั้งค่าฐานข้อมูลสำหรับการเชื่อมต่อกับฐานข้อมูล MySQL
- `set_error_msg` พุทธิตกรรมสำหรับรับข้อผิดพลาดของการทำงาน

- `set_mysql_version` พฤติกรรมสำหรับตั้งค่ารุ่นของฐานข้อมูล MySQL
- `set_resultdb` พฤติกรรมรับผลการทำงานของการเรียกใช้ฐานข้อมูล

ตัวอย่างชุดคำสั่ง `mysql_config` และ `mysql_operator` ที่สร้างขึ้น ซึ่งเป็นตัวอย่างที่ผู้เขียนใช้งานประจำ (อาจมีข้อผิดพลาดบ้าง)

```
<?php
class mysql_config {

    var $dbconfig = array(
        'hostname' => NULL ,
        'username' => NULL ,
        'password' => NULL ,
        'database' => NULL ,
        'collation_connection' => NULL ,
        'character_set' => NULL
    );

    function mysql_config(){
        $GLOBALS['global_mysql_querydb_timer'] = 0;
        $GLOBALS['global_mysql_querydb_counter'] = 0;
    }

    function get_querydb_timer() {

        global $static_query_timer;

        if(isset($static_query_timer)) return $static_query_timer;
        else return $GLOBALS['global_mysql_querydb_timer'];

    }

    function get_querydb_counter() {
        global $static_querydb_count;

        if(isset($static_querydb_count)) return $static_querydb_count;
        else return $GLOBALS['global_mysql_querydb_counter'];
    }

    function set_querydb_timer($value) {
        global $static_query_timer;

        if(isset($static_query_timer)) $static_query_timer = $value;
        else $GLOBALS['global_mysql_querydb_timer'] = $value;
    }

    function set_querydb_counter($value) {
        global $static_querydb_count;

        if(isset($static_querydb_count)) $static_querydb_count = $value;
        else $GLOBALS['global_mysql_querydb_counter'] = $value;
    }

    function set_hostname($value) {
        $this->dbconfig['hostname'] = $value;
    }

    function set_username($value) {
        $this->dbconfig['username'] = $value;
    }
}
```

```

function set_password($value) {
    $this->dbconfig['password'] = $value;
}

function set_database($value) {
    $this->dbconfig['database'] = $value;
}

function set_collation_connection($value) {
    $this->dbconfig['collation_connection'] = $value;
}

function set_character_set($value) {
    $this->dbconfig['character_set'] = $value;
}

function get_configdb() {
    return $this->dbconfig;
}
}

class mysql_operator extends mysql_config {

    var $link = NULL;

    var $result = NULL;

    var $mysql_version = NULL;

    var $string_error = array();

    function set_configdb(&$array_dbconfig) {

        $this->set_hostname($array_dbconfig['hostname']);
        $this->set_username($array_dbconfig['username']);
        $this->set_password($array_dbconfig['password']);
        $this->set_database($array_dbconfig['database']);
        $this->set_collation_connection(
            $array_dbconfig['collation_connection']
        );
        $this->set_character_set($array_dbconfig['character_set']);

    }

    function set_error_msg($string){
        array_push($this->string_error, $string);
    }

    function get_error_msg(){
        while (count($this->string_error) - 1)
            $string_output = '<b>MySQL Error : </b> '.
                array_pop($this->string_error).'<br />';
        return $string_output;
    }

    function mysql_operator(&$dbconfig) {
        if(is_array($dbconfig)) $this->set_configdb($dbconfig);
        else $this->set_configdb($dbconfig->get_configdb());
    }

    function set_mysql_version($value) {
        $this->mysql_version = $value;
    }
}

```

```

function get_mysql_version() {
    return $this->mysql_version;
}

function opendb($newlink = FALSE) {
    $this->link = mysql_connect($this->dbconfig['hostname'],
                               $this->dbconfig['username'],
                               $this->dbconfig['password'],
                               $newlink);

    if (!$this->link)
        $this->set_error_msg(
            mysql_errno($this->link).':'.
            mysql_error($this->link)
        );
    $this->set_mysql_version(mysql_get_server_info());
    if (!mysql_select_db ($this->dbconfig['database']))
        $this->set_error_msg(
            mysql_errno($this->link).':'.
            mysql_error($this->link)
        );
    if(!empty($this->dbconfig['character_set'])) {
        $this->querydb("SET CHARACTER SET ".
            $this->dbconfig['character_set'].";"
        );
        $this->querydb_counter_decrement();
    }
    if(!empty($this->dbconfig['collation_connection'])) {
        $this->querydb("SET collation_connection = ".
            $this->dbconfig['collation_connection'].";");
        $this->querydb_counter_decrement();
    }
}

function closedb() {
    if (!mysql_close($this->link))
        $this->set_error_msg(mysql_errno($this->link).':'.
            mysql_error($this->link)
        );
}

function querydb($string_query) {

    $start_time = array_sum(explode(chr(32), microtime()));
    $this->result = mysql_query ($string_query);
    $end_time = array_sum(explode(chr(32), microtime()));

    if (!$this->result)
        $this->set_error_msg(
            mysql_errno($this->link).':'.mysql_error($this->link)
        );

    $this->set_querydb_timer(
        $this->get_querydb_timer() + ($end_time - $start_time)
    );
    $this->querydb_counter_increment();
}

function list_fields($table) {
    $fields = mysql_list_fields(
        $this->dbconfig['database'], $table, $this->link
    );
}

```

```

        $columns = mysql_num_fields($fields);
        for ($i = 0; $i < $columns; $i++)
            $array_fields_in_table[$i] = mysql_field_name($fields, $i);

        return $array_fields_in_table;
    }

    function querydb_counter_increment() {
        $this->set_querydb_counter($this->get_querydb_counter()+1);
    }

    function querydb_counter_decrement() {
        $this->set_querydb_counter($this->get_querydb_counter()-1);
    }

    function querydb_transac_begin() {
        if($this->get_mysql_version() > 3)
            $this->querydb("START TRANSACTION;");
    }

    function querydb_transac_commit() {
        if($this->get_mysql_version() > 3) $this->querydb("COMMIT;");
    }

    function querydb_transac_rollback() {
        if($this->get_mysql_version() > 3) $this->querydb("ROLLBACK;");
    }

    function querydb_transac_set_autocommit_to_zero() {
        if($this->get_mysql_version() > 3)
            $this->querydb("SET AUTOCOMMIT=0;");
    }

    function querydb_transac_check_for_rollback_or_commit() {
        if(!$this->get_resultdb())$this->querydb_transac_rollback();
        else $this->querydb_transac_commit();
    }

    function num_rows_in_table($table, $condition = NULL) {
        $num_rows = new mysql_operator($this->get_configdb());
        $num_rows->opendb();
        $num_rows->querydb("SELECT COUNT(*) FROM $table $condition;");
        $num_rows->closedb();
        return mysql_result($num_rows->get_resultdb(), 0);
    }

    function has_data_in_table($table, $field, $value) {
        $value = trim($value);

        if(empty($field) or empty($value)) $sql_condtion = NULL;
        else $sql_condtion = "WHERE ".$field."='".$value.'"";

        if($this->num_rows_in_table($table, $sql_condtion) > 0) return TRUE;
        else return FALSE;
    }

    function reset_querydb() {
        $this->set_resultdb(NULL);
    }

    function set_resultdb($resource) {
        $this->result = $resource;
    }

```

```

function get_resultdb() {
    return $this->result;
}

function get_resultdb_fetch_data() {
    return mysql_fetch_array($this->get_resultdb());
}

function get_resultdb_num_rows() {
    return mysql_num_rows($this->get_resultdb());
}

function get_resultdb_num_fields() {
    return mysql_num_fields($this->get_resultdb());
}

function get_last_insert_id(){
    return mysql_insert_id();
}

function get_resultdb_atrow($row = 0) {
    return mysql_result($this->get_resultdb(), $row);
}
}
?>

```

จากตัวอย่างชุดคำสั่งนี้เราสามารถนำเข้าสู่ชุดคำสั่งที่ซับซ้อน มาทำงานได้โดยง่าย โดยเราสามารถนำไปใช้งานได้
ด้วยวิธีดังต่อไปนี้

สร้างวัตถุที่เป็นตั้งค่าการใช้งานฐานข้อมูล

```

<?php
$obj_dbconfig = new mysql_config();
$obj_dbconfig->set_hostname("hostname");
$obj_dbconfig->set_username("username");
$obj_dbconfig->set_password("password");
$obj_dbconfig->set_database("databasename");
$obj_dbconfig->set_character_set("utf8");
$obj_dbconfig->set_collation_connection("utf8_general_ci");
?>

```

สร้างวัตถุเพื่อทำงานกับฐานข้อมูล โดยเราจะทำการหับจำนวนข้อมูลในตาราง และดึงข้อมูลจากตารางที่
ชื่อว่า **table** ออกมาทั้งหมด

```

<?php
$obj_db = new mysql_operator($obj_dbconfig->get_configdb());
$obj_db->opendb();
$obj_db->querydb("SELECT * FROM table;");
$obj_db->closedb();
echo $obj_db->get_resultdb_num_rows();
while ($row = $obj_db->get_resultdb_fetch_data()) {
    echo $row[0]."<br />";
}
?>

```

จากตัวอย่างนี้ เราเพียงแต่โยนวัตถุที่มีข้อมูลตั้งค่าเพื่อติดต่อฐานข้อมูลของฐานข้อมูลในแม่แบบแล้วทำการสร้าง
วัตถุออกมาตัววัตถุนั้นก็จะมีข้อมูลตั้งค่าเพื่อติดต่อฐานข้อมูลของระบบอยู่ภายใน เมื่อเรานำไปใช้งานเพื่อทำการติดต่อ

กับฐานข้อมูลก็สามารถทำได้เลยโดยไม่ต้องกังวลว่าจะมีความผิดพลาดในการจัดจำโครงสร้างหลัง ๆ เราเพียงแต่นำเอาวัตถุที่ได้ไปใช้งานได้ตามสะดวก และทำให้ตัวชุดคำสั่งอ่านได้ง่ายมากขึ้น

จากตัวอย่างชุดคำสั่งที่ผู้เขียนนำมาใช้กับงานต่าง ๆ นั้น เรานำเอามาประยุกต์ใช้กับงานได้มากมายหลากหลายรูปแบบมาก โดยทางผู้เขียนจะนำชุดตัวอย่างโปรแกรมที่ใช้กับแม่แบบเหล่านี้มาให้ดาวน์โหลดในเว็บ

<http://www.thaicyperpoint.com/ford/blog/projects> โดยจะอยู่ใน Section ของฐานข้อมูลครับ รวมถึงโปรแกรมที่ผู้เขียนได้นำเอาแม่แบบ `mysql_config` และ `mysql_operator` ไปใช้งานบางส่วนมาเปิดให้ดาวน์โหลดกันด้วย โดยที่ตัวอย่างดังกล่าวจะมีการใช้งานทุก ๆ พฤติกรรมอย่างสมบูรณ์กว่าในบทความนี้ (ด้วยข้อจำกัดด้านจำนวนหน้าในบทความ)

ในตอนนี้ตัวชุดคำสั่งโปรแกรมต่าง ๆ อาจจะดูเยอะหน่อย ด้วยเหตุผลที่ว่าอยากนำไปสังเกตกับการใช้งานจริงว่าเรานำเอาไปประยุกต์ใช้กับงานได้หลากหลาย และมีการเข้าถึงตัวข้อมูลของวัตถุได้อย่างไรบ้าง

ท่านผู้อ่านท่านใดสงสัยประการใดสามารถอีเมลมาได้ที่ annop@thaicyberpoint.com ได้ทุกเมื่อครับ

เอกสารอ้างอิง

- **PHP Manual**, <http://www.php.net>, January 2007
- **Zend PHP Certification Study Guide**, Zend Technologies, August 2004
- **Concepts of Programming Languages (7th Edition)**, Robert W. Sebesta, April 2005