Chapter 2

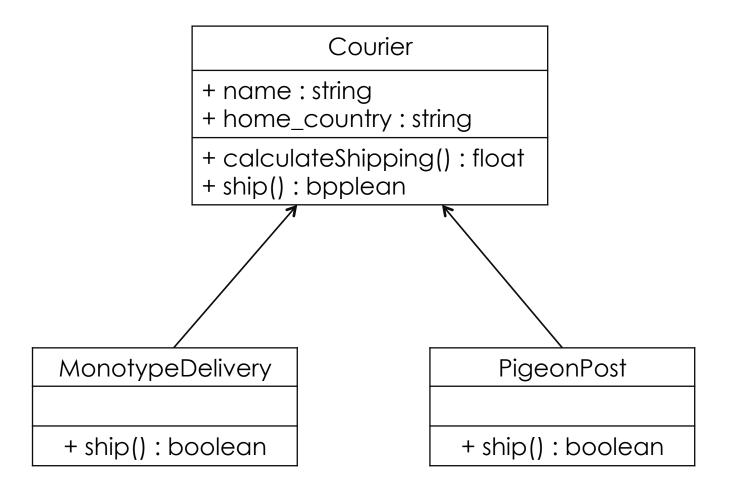
# PHP

53363393 Juthathip Kaensing Computer Engineering Naresuan University

## การเขียนโปรแกรม เชิงใจกัด

## การสืบทอดของ Object

#### การสืบทอดจากคลาส Courier



```
class Courier
   public $name;
   public $home_country;
   public function __construct($name, $home_country) {
        $this->name = $name;
        return true;
   public function ship($parcel) {
        // sends the parcel to its destination
        return true;
   public function calculateShipping($parcel) {
        // look up the rate for the destination, we'll invent one
        rate = 1.78;
        // calculate the cost
        $cost = $rate * $parcel->weight;
        return $cost;
```

```
class MonotypeDelivery extends Courier
{
  public function ship($parcel) {
    // put in box
    // send
    return true;
  }
}
```

```
class PigeonPost extends Courier
{
  public function ship($parcel) {
    // fetch pigeon
    // attach parcel
    // send
    return true;
  }
}
```

### Object และ Function

```
class PigeonPost extends Courier
{
  public function ship($parcel) {
     // fetch pigeon
     // attach parcel
     // send
     return true;
  }
}
```

## Type Hinting

```
public function ship(Parcel $parcel) {
    // sends the parcel to its destination
    return true;
}
```

## Polymorphism

```
function saveAsPreFerredSupplier(Courier $courier) {
    // add to list and save
    return true;
}
```

```
$courier = new PigeonPost('Local Avian Delivery Ltd');
if($courier instanceOf Courier) {
    echo $couier->name , "is a Courier \n";
}
if($courier instanceOf PigeonPost) {
    echo $courier->name , "is a PigeonPost \n"
}
if($courier instanceOf Parcel) {
    echo $courier->name , "is a Parcel \n"
}
```

Local Avian Delivery Ltd is a Courier Local Avian Delivery Ltd is a PigeonPost

Chapter 2

### Object และ Reference

```
$box1 = new Parcel();
$box1->destinationCountry = 'Denmark';

$box2 = new $box1();
$box2->destinationCountry = 'Brazil';

echo 'Parcels need to ship to: '
    , $box1->destinationCountry, ' and '
    , $box2->destinationCountry;
}
```

Parcels need to ship to: Brazil and Brazil

#### Chapter 2

```
if($box1 == $box2) echo 'equivalent';
```

```
if($box1 === $box2) echo 'exact same object'
```

### การส่งผ่านออบเจ็กต์ ให้เป็นพารามิเตอร์**ของ**ฟังก์ชั่น

```
$courier = new PigeonPost('Avian Delivery Ltd');
$other_courier = $courier;
$other_courier->name = 'Pigeon Post';
echo $courier->name; // outputs "Pigeon Post"
```

```
$courier = new PigeonPost('Avian Delivery Ltd');
$other_courier = clone $courier;
$other_courier->name = 'Pigeon Post';
echo $courier->name; // outputs "Avian Delivery Ltd"
```

### Fluent Interface

#### Chapter 2

```
● ● ●
                                  Parcel.php
            Parcel.php > No Selection
   <?php
   class Parcel
     public $weight;
     public $destinationAddress;
     public $destinationCountry;
     public function setWeight($weight) {
       echo "weight set to: " . $weight . "\n";
       $this->weight = $weight;
       return $this;
     public function setCountry($country) {
       echo "destination country is: " . $country . "\n";
       $this->destinationCountry = $country;
       return $this;
   } //end of Parcel class
   $myparcel = new Parcel();
   $myparcel->setWeight(5)->setCountry('Peru');
   ?>
```

## Public, Private และ Protected

Chapter 2





Chapter 2



### การใช้ Getter และ Setter เพื่อ ควบคุมการมองเห็น

```
class Courier
{
    protected $name;

    function getName() {
        return $this->name;
    }

    function setName($value) {
        $this->name = $value;
        return true;
    }
}
```

การใช้ method \_\_get และ \_\_set

```
class Courier
{
    protected $data = array();

    public function __get($property) {
        return $this->data[$property];
    }

    public function __set($property, $value) {
        $this->data[$property] = $value;
        return true;
    }
}
```

```
$courier = new Courier();
$courier->name = 'Avian Carrier';
echo $courier->name;
```

## interface

#### ตัวอย่างการใช้ interface Countable

```
class Courier implements Countable
{
  protected $count = 0;

  public function ship(Parcel $parcel) {
    $this->count++;
    // ship parcel
    return true;
  }

  public function count() {
    return $this->count;
  }
}
```

## Counting Object

```
$courier = new Courier();
$courier->ship(new Parcel());
$courier->ship(new Parcel());
$courier->ship(new Parcel());
$courier->ship(new Parcel());
echo count($courier); //outputs 3
```

#### การประกาศและการใช้งาน interface class

```
interface Trackable
{
   public function getTrackInfo($parcelId);
}
```

```
class MonotypeDelivery extends Courier implements Trackable
{
    public function ship($parcel) {
        // put in box
        // send and get parcel ID (we'll just pretend)
        $parcelId = 42;
        return $parcelId;
    }

    public function getTrackInfo($parcelId) {
        // look up some information
        return(array("status" => "in transit"));
    }
}
```

#### การระบุตัวตน Object และ interface

```
$courier = new MonotypeDelivery();
if($courier instanceOf Courier) {
    echo "I'm a Courier\n";
if($courier instanceOf MonotypeDelivery) {
    echo "I'm a MonotypeDelivery\n";
if($courier instanceOf Parcel) {
    echo "I'm a Parcel\n";
if($courier instanceOf Trackable) {
    echo "I'm a Trackable\n":
/*
Output;
I'm a Courier
I'm a MonotypeDelivery
 I'm a Trackable
*/
```

## Exception

### การจัดการกับ Exception

```
$db = new PDO('mysql:host=nonsense');
```

```
try {
    $db = new PDO('mysql:host=nonsense')
    echo "Connected to database";
}
    catch (Exception $e){
    echo "Oops! " . $e->getMessage();
}
```

## ไม่มี Finally blog

Try/catch/finally

## ทำไมต้อง Exception

## Throw exception

```
//something has hone wrong
throw new Exception('Meaningful error message string');
```

### การ extends Exception

class HeavyParcelException extends Exception {}

#### การจับ Exception ที่จำเพาะเจาะจงประเภท

```
HeavyParcelException.php
       HeavyParcelException.php > No Selection
<?php
class HeavyParcelException extends Exception {}
class Courier{
  public function ship(Parcel $parcel) {
    // check we have an address
    if(empty($parcel->address)) {
      throw new Exception('Address not Specified');
    // check the weight
    if($parcel->weight > 5) {
      throw new HeavyParcelException('Parcel exceeds courier
          limit');
    // otherwise we're cool
    return true;
```

```
$myCourier = new Courier();
$parcel = new Parcel();
// add the address if we have it
parcel->weight = rand(1,7);
try {
    $myCourier->ship($parcel);
    echo "parcel shipped";
catch (HeavyParcelException $e) {
    echo "Parcel weight error: " . $e->getMessage();
    // redirect them to choose another courier
catch (Exception $e) {
    echo "Something went wrong. " . $e->getMessage();
    // exit so we don't try to proceed any further
   exit:
```

"Parcel shipped"

Uncaught Exception

### การกำหนด Global Exception

#### Set\_exception\_handler()

```
function handleMissedException($e) {
    echo "Sorry, something is wrong. Please try again, or contact us
        if the problem persists";
    error_log('Unhandled Exception: ' . $e->getMessage(). ' in file '
        . $e->getFile() . ' on line ' . $e->getLine());
}
```

```
set_exception_handler('handleMissedException');
throw new Exception('just testing!');
```

### Error\_log

[13-Jan-2012 11:25:41] Unhandled Exception: just testing! in file /
home/lorna/.../exception-handler.php on line 13

### Callback

- ชื่อฟังก์ชั่น
- o ชื่อคลาสและเมธอดที่ถูกเรียกแบบ static
- ชื่อออบเจ็กต์และชื่อเมธอดที่เรียกมาจากออบเจ็กต์ ต้นทาง
- โครเชอร์ (ฟังก์ชั่นที่ถูกเก็บไว้ในตัวแปร)
- แลมด้าฟังก์ชั่น (ฟังก์ชั่นที่ถูกสร้างขึ้นขณะที่ต้องการ ใช้)

## สรุปเรื่องเมจิกเมธอด

| ฟังก์ชั่น    | ทำงานเมื่อ                    |
|--------------|-------------------------------|
| _construct() | Object ถูกจำลอง               |
| _destruct()  | Object ถูกทำลาย               |
| _get()       | Property ที่ไม่มีอยู่ถูกอ่าน  |
| _set()       | Property ที่ไม่มีอยู่ถูกเขียน |
| _clone()     | Object ถูก copy               |

การใช้งาน \_\_call() และ \_\_callStatic()

```
class Courier
    public $name;
    public function __construct($name) {
        $this->name = $name;
        return true;
    public function ship($parcel) {
        // sends the parcel to its destination
        return true;
    public function __call($name, $params) {
        if($name == 'sendParcel') {
            // legacy system requirement, pass to newer send() method
            return $this->send($params[0]);
        } else {
            error_log('Failed call to ' . $name . ' in Courier class');
                return false:
```