Lập trình php2

Php oop 3

⦿ OOP Error Handling

⦿ Datetime Object OOP

⦿ PHP Superglobals

⦿ PHP OOP và MVC Pattern

PHẦN 1

❏Throwing Exceptions: sử dụng keyword `throw`

```php
class Customer{
  public function __construct(private $billingInfo = [])
  {
  }
  public function getBillingInfo():array {
    return $this->billingInfo;
  }
}
class Invoice {
  public function __construct(public Customer $customer)
  {
  }
  public function process(float $amount): void{
    if($amount < 0){
      throw new \Exception('Invalid invoice amount');
    }
    echo 'Processing $'.$amount.' invoice ';
    sleep(1);
    echo 'OK'.PHP_EOL;
  }
}
```

```php
$invoice = new Invoice(new Customer());
$invoice->process(25);
```

output

PROBLEMS   OUTPUT   DEBUG CONSOLE

```
TramnoMBookpuro:trash tram$ php example.php
Processing $25 invoice OK
TramnoMBookpuro:trash tram$ 
```

❑**Throwing Exceptions:** sử dụng keyword `throw`

```php
class Customer{
  public function __construct(private $billingInfo = [])
  {
  }
  public function getBillingInfo():array {
    return $this->billingInfo;
  }
}
class Invoice {
  public function __construct(public Customer $customer)
  {
  }
  public function process(float $amount): void{
    if($amount < 0){
      throw new \Exception('Invalid invoice amount');
    }
    echo 'Processing $'.$amount.' invoice ';
    sleep(1);
    echo 'OK'.PHP_EOL;
  }
}
```

exception

```php
$invoice = new Invoice(new Customer());
$invoice->process(-25);
```

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE   🔲 bash - trash ✚ ∨ ▢ 🗑

```
TramnoMBookpuro:trash tram$ php example.php
PHP Fatal error:  Uncaught Exception: Invalid invoice amount in /Users/tram/.bit
i/stackman/machines/xampp/volumes/root/htdocs/PHP2/mvc_simple/trash/example.php:
Stack trace:
#0 /Users/tram/.bitnami/stackman/machines/xampp/volumes/root/htdocs/PHP2/mvc_sim
/trash/example.php(25): Invoice->process(-25)
#1 {main}
  thrown in /Users/tram/.bitnami/stackman/machines/xampp/volumes/root/htdocs/PHP
vc_simple/trash/example.php on line 16

Fatal error: Uncaught Exception: Invalid invoice amount in /Users/tram/.bitnami/
```

```php
class Customer{
  public function __construct(private $billingInfo = [])
  {
  }
  public function getBillingInfo():array {
    return $this->billingInfo;
  }
}
class Invoice {
  public function __construct(public Customer $customer)
  {
  }
  public function process(float $amount): void{
    if($amount < 0){
      throw new \Exception('Invalid invoice amount');
    }
    if(empty($this->customer->getBillingInfo())){
      throw new MissingBillingInfoException();
    }
    echo 'Processing $'.$amount.' invoice ';
    sleep(1);
    echo 'OK'.PHP_EOL;
  }
}
```

Example

```php
try{
  $invoice->process(25);
}catch (MissingBillingInfoException){
  echo 'Some error';
}
```

Call →

```php
class MissingBillingInfoException extends \Exception{
  protected $message = 'Missing billing information';
}
```

```php
class Customer{
  public function __construct(private $billingInfo = [])
  {
  }
  public function getBillingInfo():array {
    return $this->billingInfo;
  }
}
class Invoice {
  public function __construct(public Customer $customer)
  {
  }
  public function process(float $amount): void{
    if($amount < 0){
      throw new \Exception('Invalid invoice amount');
    }
    if(empty($this->customer->getBillingInfo())){
      throw new MissingBillingInfoException();
    }
    echo 'Processing $'.$amount.' invoice ';
    sleep(1);
    echo 'OK'.PHP_EOL;
  }
}
```

Example

```php
try{
  $invoice->process(25);
}catch (MissingBillingInfoException){
  echo 'Some error';
}catch(\InvalidArgumentException){
  echo 'Invalid argument exception'.PHP_EOL;
}
```

Call

```php
class MissingBillingInfoException extends \Exception{
  protected $message = 'Missing billing information';
}
```

```php
class Customer{
  public function __construct(private $billingInfo = [])
  {
  }
  public function getBillingInfo():array {
    return $this->billingInfo;
  }
}
class Invoice {
  public function __construct(public Customer $customer)
  {
  }
  public function process(float $amount): void{
    if($amount < 0){
      throw new \Exception('Invalid invoice amount');
    }
    if(empty($this->customer->getBillingInfo())){
      throw new MissingBillingInfoException();
    }
    echo 'Processing $'.$amount.' invoice ';
    sleep(1);
    echo 'OK'.PHP_EOL;
  }
}
```

Example

```php
try{
  $invoice->process(-25);
}catch(\MissingBillingInfoException|\InvalidArgumentException $e){
  echo $e->getMessage().PHP_EOL;
}
```

or

Call

```php
class MissingBillingInfoException extends \Exception{
  protected $message = 'Missing billing information';
}
```

```php
class Customer{
  public function __construct(private $billingInfo = [])
  {
  }
  public function getBillingInfo():array {
    return $this->billingInfo;
  }
}
class Invoice {
  public function __construct(public Customer $customer)
  {
  }
  public function process(float $amount): void{
    if($amount < 0){
      throw new \Exception('Invalid invoice amount');
    }
    if(empty($this->customer->getBillingInfo())){
      throw new MissingBillingInfoException();
    }
    echo 'Processing $'.$amount.' invoice ';
    sleep(1);
    echo 'OK'.PHP_EOL;
  }
}
}
```

Example

```php
try{
  $invoice->process(-25);
}catch(\Exception $e){
  echo $e->getMessage().PHP_EOL;
}finally{
  echo 'Finally block'.PHP_EOL;
}
```

Call

```php
class MissingBillingInfoException extends \Exception{
  protected $message = 'Missing billing information';
}
```

tommorrow

next tuesday

last day of month

tommorrow noon

10/10/2021 7:00

………

```php
$datetime = new DateTime();
var_dump($datetime)
```

Output

```
object(DateTime)#1 (3) {
  ["date"]=>
  string(26) "2021-12-28 18:14:21.904135"
  ["timezone_type"]=>
  int(3)
  ["timezone"]=>
  string(3) "UTC"
}
```

tommorrow

next tuesday

last day of month

tommorrow noon

10/10/2021 7:00

....

```php
$datetime = new DateTime('tomorrow 3:35pm');
var_dump($datetime)
```

Output

```
object(DateTime)#1 (3) {
  ["date"]=>
  string(26) "2021-12-29 15:35:00.000000"
  ["timezone_type"]=>
  int(3)
  ["timezone"]=>
  string(3) "UTC"
}
```

tommorrow

next tuesday

last day of month

tommorrow noon

10/10/2021 7:00

………

```php
$datetime = new DateTime('yesterday noon');
var_dump($datetime)
```

Output

```
object(DateTime)#1 (3) {
  ["date"]=>
  string(26) "2021-12-27 12:00:00.000000"
  ["timezone_type"]=>
  int(3)
  ["timezone"]=>
  string(3) "UTC"
}
```

tommorrow

next tuesday

last day of month

tommorrow noon

10/10/2021 7:00

………

```php
$datetime = new DateTime('5/12/2021 3:30PM');

echo $datetime->getTimezone()->getName().' – '.$datetime->format('m/d/Y g:i A').PHP_EOL;

$datetime->setDate(2021,4,21)->setTime(2,15);
$datetime->setTimezone(new DateTimeZone('Europe/Amsterdam'));

echo $datetime->getTimezone()->getName().' – '.$datetime->format('m/d/Y g:i A').PHP_EOL;
```

Output

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

```
TramnoMBookpuro:trash tram$ php example.php
UTC – 05/12/2021 3:30 PM
Europe/Amsterdam – 04/21/2021 4:15 AM
```

compare

tommorrow

next tuesday

last day of month

tommorrow noon

10/10/2021 7:00

………

```php
$date = '15/05/2021';

$datetime = DateTime::createFromFormat('d/m/Y',$date)->setTime(0,0);

var_dump($datetime, new DateTime('15-05-2021'));

$dateTime1 = new DateTime('05/25/2021 9:15 AM');
$dateTime2 = new DateTime('05/25/2021 9:14 AM');

var_dump($dateTime1 < $dateTime2);
var_dump($dateTime1 > $dateTime2);
var_dump($dateTime1 == $dateTime2);
var_dump($dateTime1 <=> $dateTime2);
```
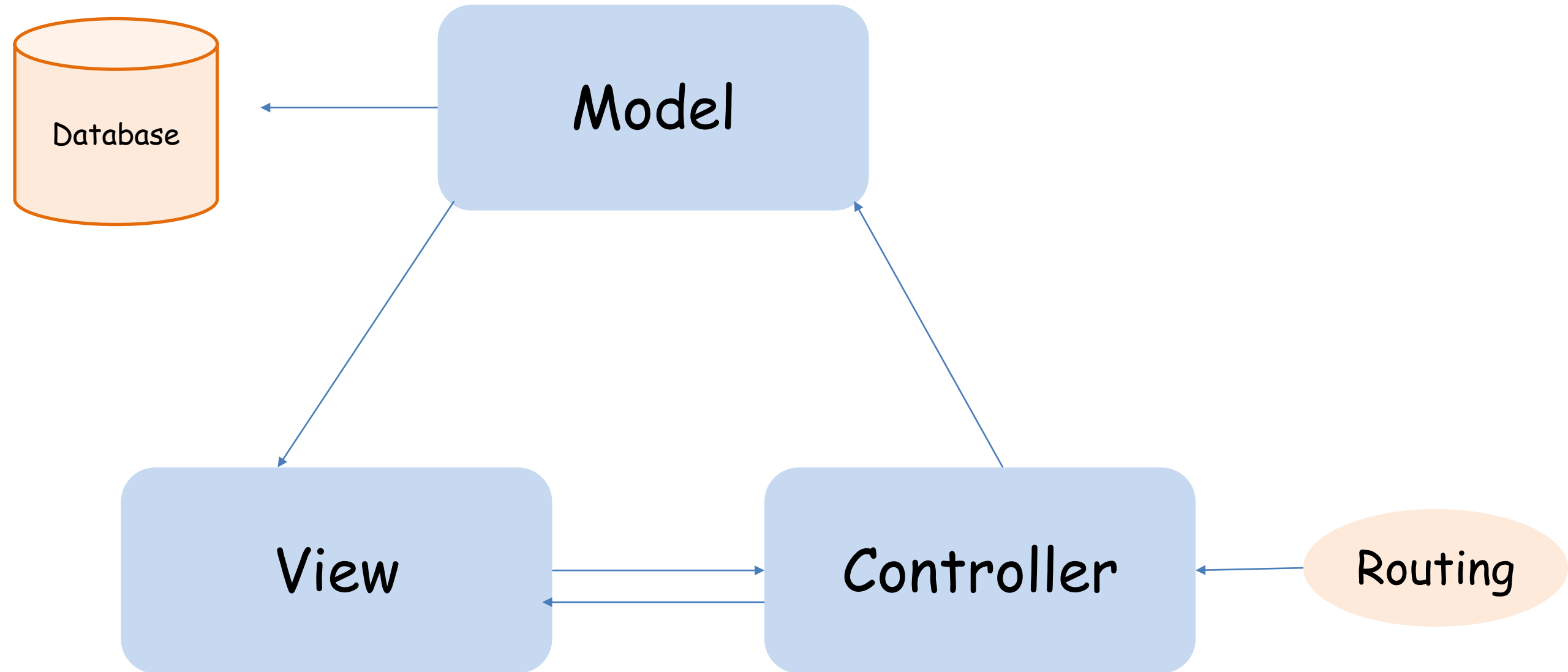
- ❑ $dateTime1->diff($dateTime2)
- ❑ new DateInterval()
- ❑ $datetime->add()
- ❑ $datetime->sub()
- ❑ new DatePeriod()

PHẦN 2

- ❑ $_SERVER
- ❑ $_GET
- ❑ $_POST
- ❑ $_FILES
- ❑ $_COOKIE
- ❑ $_SESSION
- ❑ $_REQUEST

```
Array
(
    [UNIQUE_ID] => YcvDf3h-SSC8o91UffMZOQAAAAM
    [HTTP_HOST] => 192.168.64.2
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_COOKIE] => PHPSESSID=4ohsi50qhmpdbu79pmlrgrttq0
    [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/60
    [HTTP_ACCEPT_LANGUAGE] => en-us
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_CONNECTION] => keep-alive
    [PATH] => /opt/lampp/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
    [LD_LIBRARY_PATH] => /opt/lampp/lib:/opt/lampp/lib
    [SERVER_SIGNATURE] =>
    [SERVER_SOFTWARE] => Apache/2.4.48 (Unix) OpenSSL/1.1.1k PHP/8.0.8
    [SERVER_NAME] => 192.168.64.2
    [SERVER_ADDR] => 192.168.64.2
    [SERVER_PORT] => 80
    [REMOTE_ADDR] => 192.168.64.1
    [DOCUMENT_ROOT] => /opt/lampp/htdocs
    [REQUEST_SCHEME] => http
    [CONTEXT_PREFIX] =>
    [CONTEXT_DOCUMENT_ROOT] => /opt/lampp/htdocs
    [SERVER_ADMIN] => you@example.com
    [SCRIPT_FILENAME] => /opt/lampp/htdocs/PHP2/mvc_simple/route.php
    [REMOTE_PORT] => 52863
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /PHP2/mvc_simple/route.php
    [SCRIPT_NAME] => /PHP2/mvc_simple/route.php
    [PHP_SELF] => /PHP2/mvc_simple/route.php
    [REQUEST_TIME_FLOAT] => 1640743807.2857
    [REQUEST_TIME] => 1640743807
)
```

Index.php

```php
require_once __DIR__.'/vendor/autoload.php';
use app\core\Route as Router;

$router = new Router();
$router->register(
    '/',
    function(){
        echo "Home";
    }
);
```

$route

$action

Terminal: php -S localhost:8080

```
Array
(
    [UNIQUE_ID] => YcvDf3h-SSC8o91UffMZOQAAAAM
    [HTTP_HOST] => 192.168.64.2
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xm
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_COOKIE] => PHPSESSID=4ohsi50qhmpdbu79pmlrgrttq0
    [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; Intel Mac OS X 10_
    [HTTP_ACCEPT_LANGUAGE] => en-us
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_CONNECTION] => keep-alive
    [PATH] => /opt/lampp/bin:/usr/local/sbin:/usr/local/bin:/usr/sb
    [LD_LIBRARY_PATH] => /opt/lampp/lib:/opt/lampp/lib
    [SERVER_SIGNATURE] =>
    [SERVER_SOFTWARE] => Apache/2.4.48 (Unix) OpenSSL/1.1.1k PHP/8.
    [SERVER_NAME] => 192.168.64.2
    [SERVER_ADDR] => 192.168.64.2
    [SERVER_PORT] => 80
    [REMOTE_ADDR] => 192.168.64.1
    [DOCUMENT_ROOT] => /opt/lampp/htdocs
    [REQUEST_SCHEME] => http
    [CONTEXT_PREFIX] =>
    [CONTEXT_DOCUMENT_ROOT] => /opt/lampp/htdocs
    [SERVER_ADMIN] => you@example.com
    [SCRIPT_FILENAME] => /opt/lampp/htdocs/PHP2/mvc_simple/route.ph
    [REMOTE_PORT] => 52863
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /PHP2/mvc_simple/route.php
    [SCRIPT_NAME] => /PHP2/mvc_simple/route.php
    [PHP_SELF] => /PHP2/mvc_simple/route.php
    [REQUEST_TIME_FLOAT] => 1640743807.2857
    [REQUEST_TIME] => 1640743807
)
```

Index.php

Class Route

$action

```php
require_once __DIR__.'/vendor/autoload.php';
use app\core\Route as Router;


$router = new Router();
$router->register(
  '/',
  function(){
    echo "Home";
  }
);
```

```php
protected array $routes;

public function register(string $route, callable|array $action): self
{
    //var_dump($route);
    $this->routes[$route] = $action;
    //var_dump($this->routes);
    return $this;
}
```

2

2

1

1

1

$route

Index.php

Class Route

```php
$router = new Router();
$router->register(
  '/',
  function(){
    echo "Home";
  }
);


$router->register(
  '/invoices',
  function(){
    echo "invoices";
  }
);


$router->register(
  '/invoices/create',
  function(){
    echo "Create invoices";
  }
);

echo $router->resolve($_SERVER['REQUEST_URI']);
```

```php
public function resolve(string $requestUrl){
    $route = explode('?',$requestUrl)[0];
    $action = $this->routes[$route] ?? null;
    if(!$action){
        throw new RouteNotFoundException();
    }


    if(is_callable($action)){
        return call_user_func($action);
    }


    if(is_array($action)){
        [$class, $method] = $action;
        // invoices/create
        if(class_exists($class)){
            $class = new $class();

            if(method_exists($class,$method)){
                return call_user_func_array([$class,$method],[]);
            }
        }
    }
}
```

Index.php

Class Home

Call > class: Home >
Method function: index

```php
class Home {
    public static function index(): string{
        return 'Home';
    }
}
```

```php
$router = new Router();
$router
  ->register('/', [app\Home::class,'index'])
  ->register('/invoices', [app\Invoices::class,'index'])
  ->register('/invoices/create', [app\Invoices::class,'create']);
```

Class Invoices

Call > class: ? >
Method function: ?

```php
class Invoices {
    public static function index(): string{
        return 'Invoices';
    }

    public static function create(): string{
        return 'Create Invoice';
    }
}
```

☑Inheritance

☑Access level & Class constant

☑Interface

☑Abstract

☑Trait

thank you!