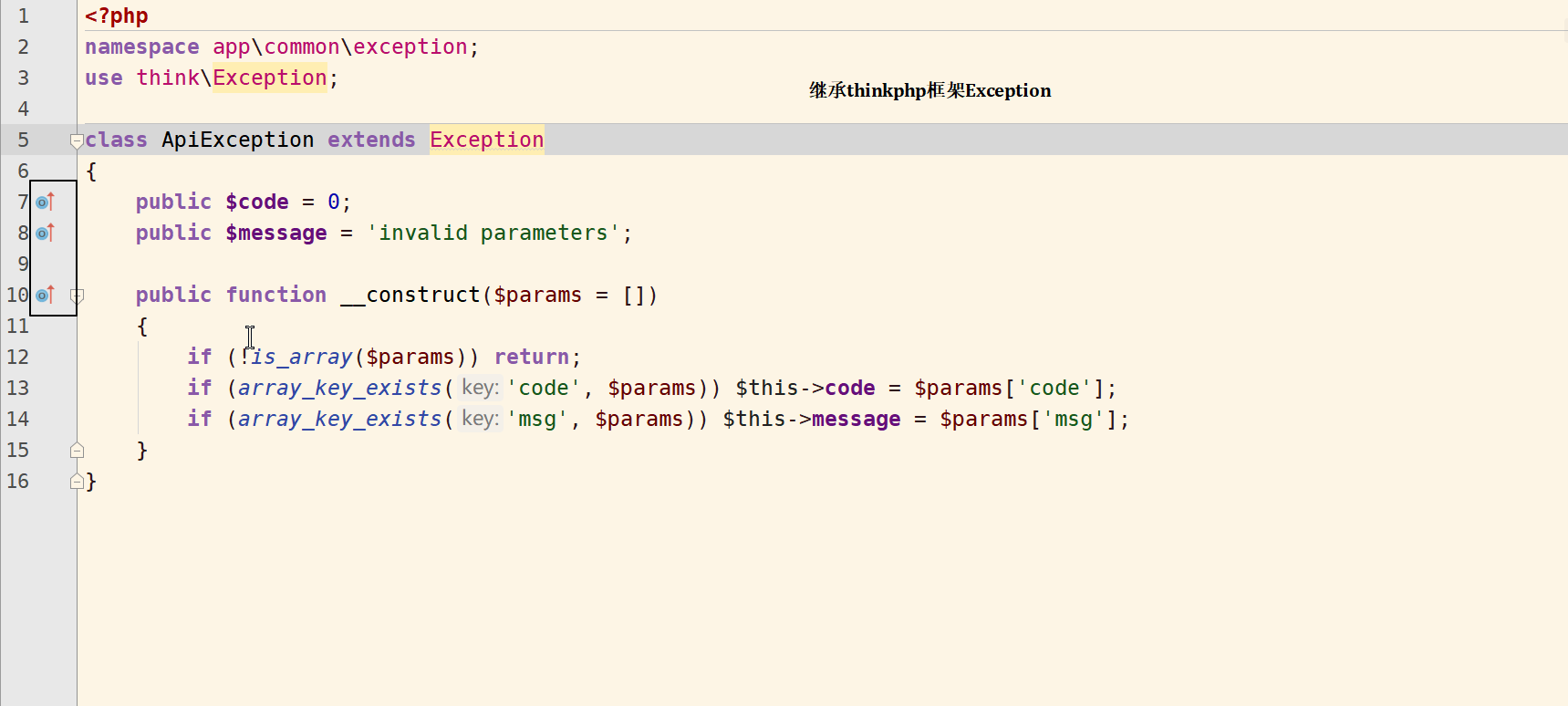
# [TP5异常处理](https://www.cnblogs.com/yanweifeng/p/10950424.html)

## **自定义异常处理**



namespace app\common\exception;use think\Exception;

class ApiException extends Exception{

public $code = 0;

public $message = 'invalid parameters';

public function \_\_construct($params = [])

{

if (!is\_array($params)) return;

if (array\_key\_exists('code', $params)) $this->code = $params['code'];

if (array\_key\_exists('msg', $params)) $this->message = $params['msg'];

}

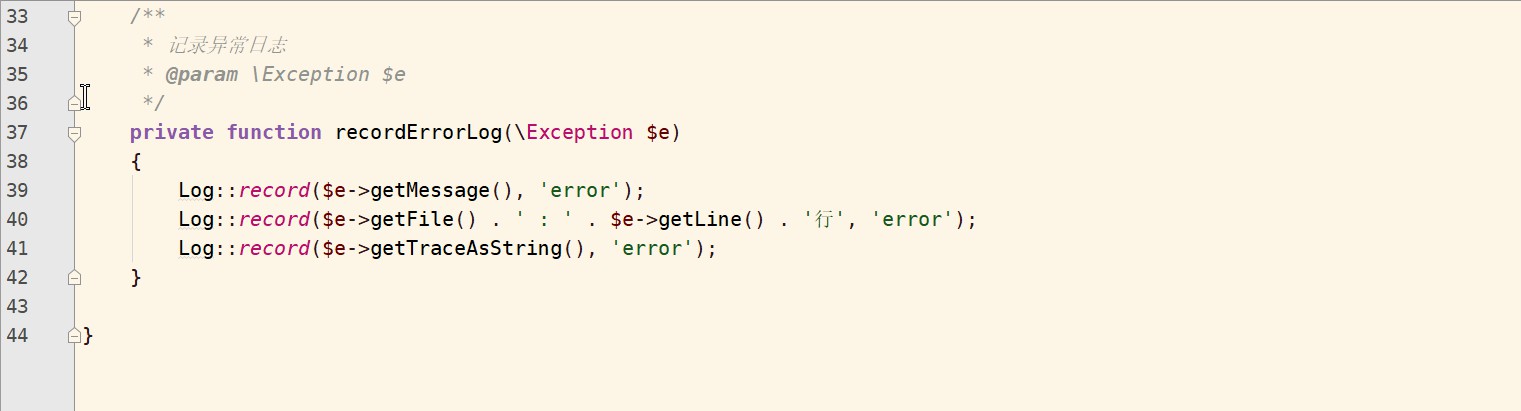
}

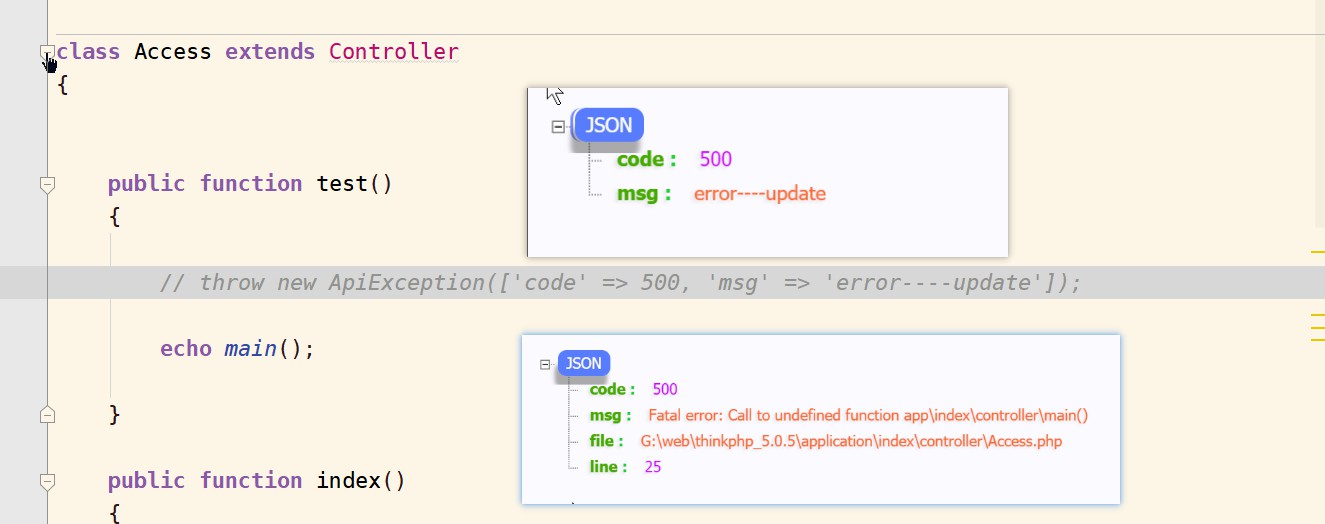
## **异常处理接管[重写]**











<?php/\*\*

\* Created by PhpStorm.

\* User: ywf

\* Date: 2019/5/16

\* Time: 11:13

\*/namespace app\common\exception;use think\Log;

class MyHttpException extends \think\exception\Handle{

public function render(\Exception $e)

{

if (config('app\_debug')) return parent::render($e);

if ($e instanceof ApiException) {

return json([

'code' => $e->code,

'msg' => $e->message,

]);

}

$this->recordErrorLog($e);

return json([

'code' => 500,

'msg' => $e->getMessage() ? : 'sorry server internal error!',

'file' => $e->getFile(),

'line' => $e->getLine(),

]);

}

/\*\*

\* 记录异常日志

\* @param \Exception $e

\*/

private function recordErrorLog(\Exception $e)

{

Log::record($e->getMessage(), 'error');

Log::record($e->getFile() . ' : ' . $e->getLine() . '行', 'error');

Log::record($e->getTraceAsString(), 'error');

}

}