PHP面向对象之封装

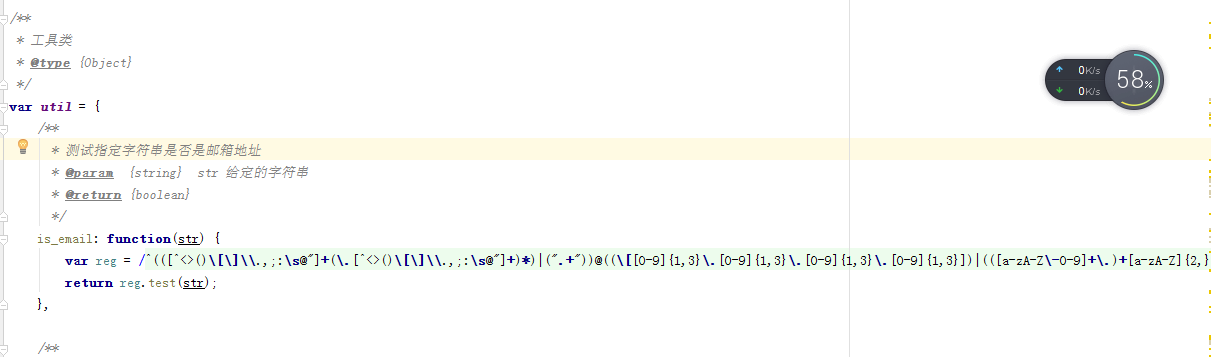
1. 什么是封装？

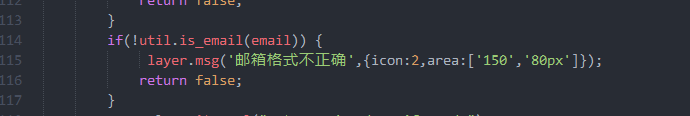
封装就是把抽取出来的数据和对数据的操作封装在一起，数据被保护在内部，程序的其他部分只有被授权的操作（方法）才能对数据进行操作。   
php提供了三种访问控制修饰符   
1. public 表示全局，本类内部，类外部，子类都可以访问   
2. protected 表示受保护的，只有本类或子类可以访问   
3. private 表示私有的，只有本类内部可以访问

<?php   
class Person{   
public $name;   
protected $age;   
private $salary;   
function \_\_construct($name,$age,$salary){   
$this->name=$name;   
$this->age=$age;   
$this->salary=$salary;   
}   
public function showinfo(){   
//这表示三个修饰符都可以在本类内部使用   
echo $this->name."||".$this->age."||".$this->salary;   
}   
}   
$p1=new Person('张三',20,3000);   
//这里属于类外部，那么如果用下面的方法访问age和salary都会报错   
// echo $p1->age; echo$p1->salary;   
?>

1. 封装的好处
2. 就是调用的时候方便，可以多个地方调用，免得重复的代码的写得很多
3. 增加了安全性，维护性。

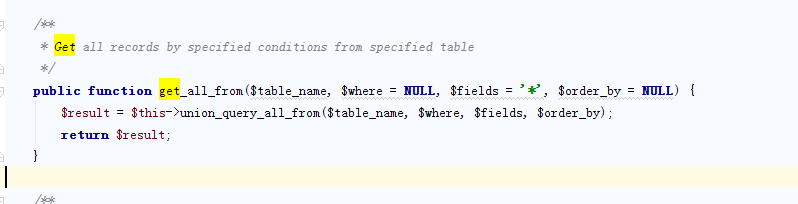
例如在:EDK项目中

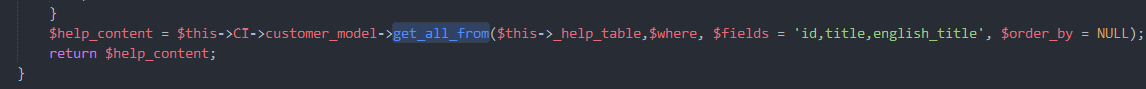




1. 这个工具类，将邮箱验证的方法进行了封装，is\_email这个方法，需要传递的参数也说得很清楚，需要传递参数是一个字符串，当在页面需要用到js验证时，只需要调用这个类下的方法，传递需要的参数，以方便在其它的地方多次调用，也就说明了封装避免的代码的重复性(单一功能的封装函数)。

2.在EDK项目 后台方法的封装





后台列表查询数据，经常调用查询的方法，如果不将其方法封装，将多次出现相同的代码，这样代码的重复性就很高，这样将其方法封装，传递此方法需要的参数，就可以在多个地方调用，减少了代码的重复性。