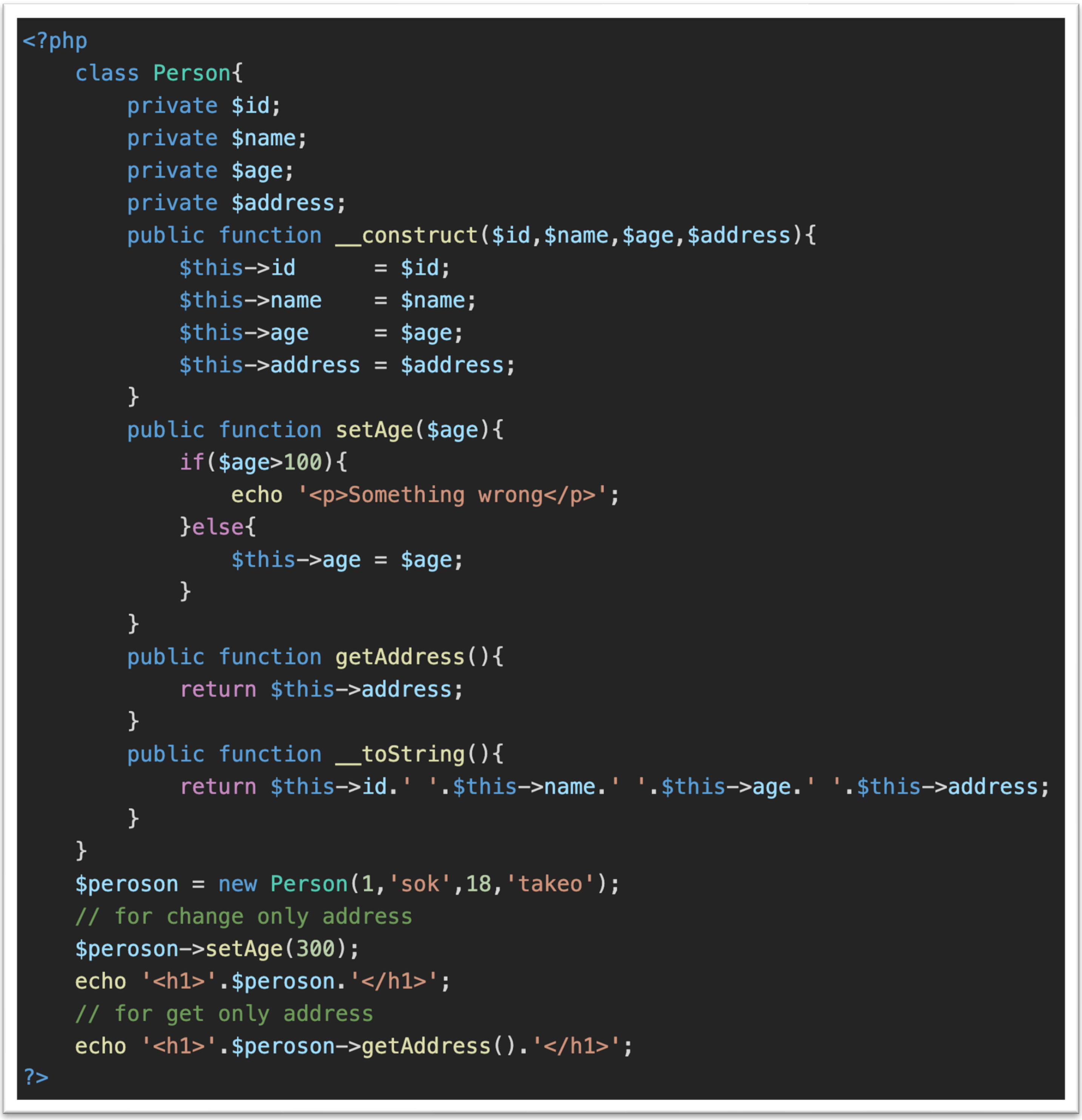
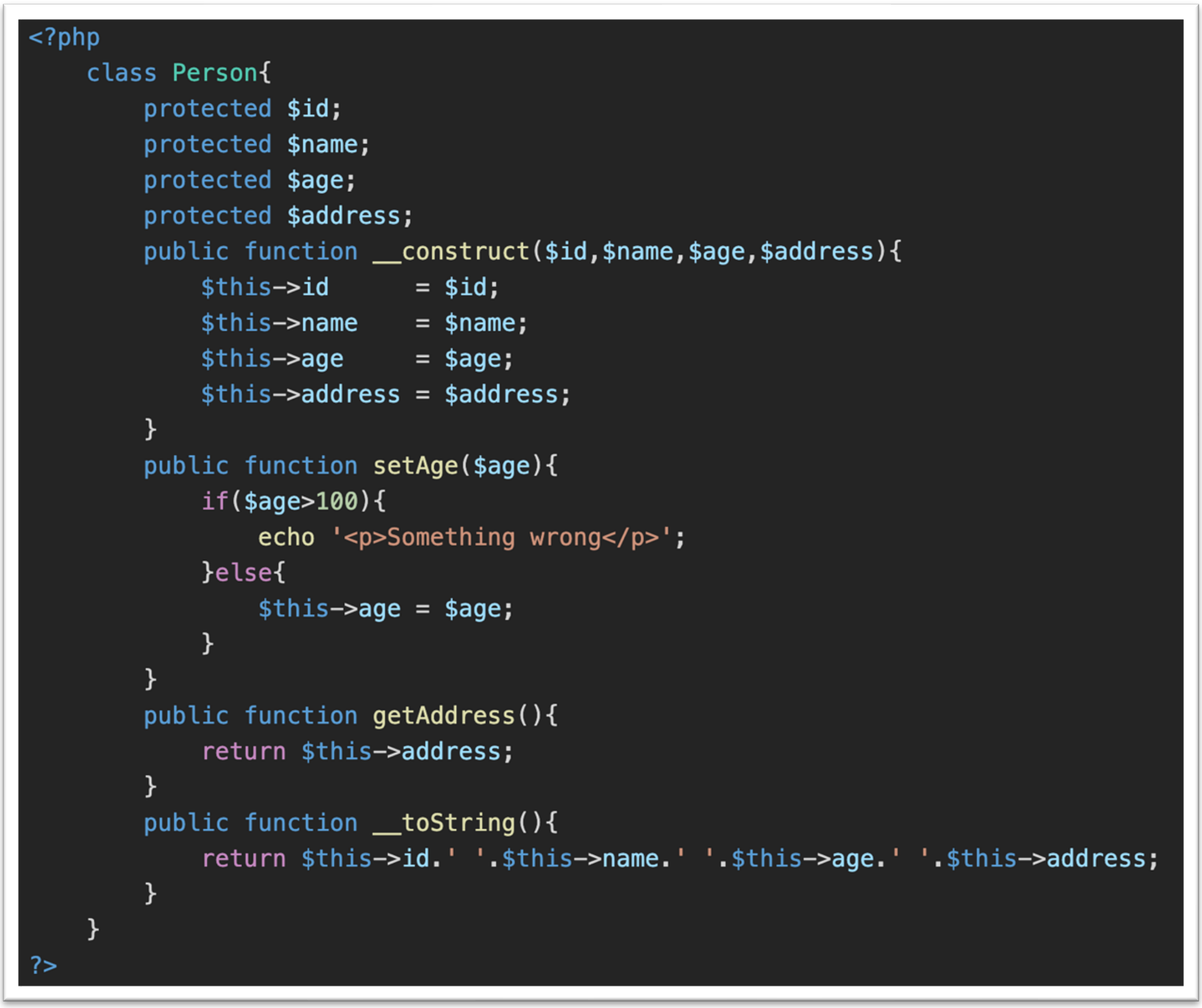
Luon Verak

+ Abstraction



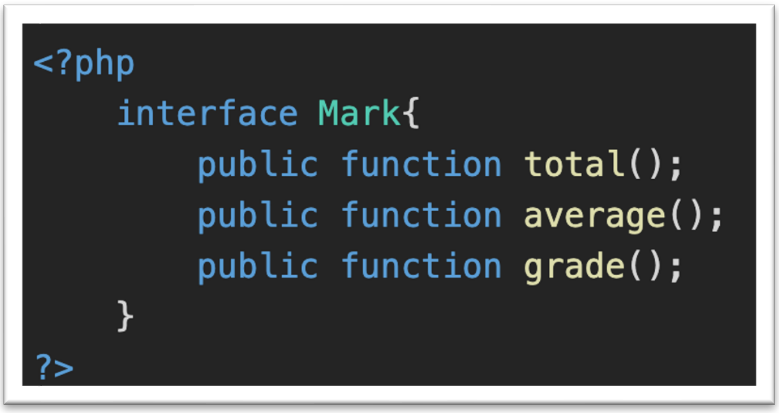
```php
<?php
    class Person{
        private $id;
        private $name;
        private $age;
        private $address;
        public function __construct($id,$name,$age,$address){
            $this->id      = $id;
            $this->name    = $name;
            $this->age     = $age;
            $this->address = $address;
        }
        public function setAge($age){
            if($age>100){
                echo '<p>Something wrong</p>';
            }else{
                $this->age = $age;
            }
        }
        public function getAddress(){
            return $this->address;
        }
        public function __toString(){
            return $this->id.' '.$this->name.' '.$this->age.' '.$this->address;
        }
    }
    $peroson = new Person(1,'sok',18,'takeo');
    // for change only address
    $peroson->setAge(300);
    echo '<h1>'.$peroson.'</h1>';
    // for get only address
    echo '<h1>'.$peroson->getAddress().'</h1>';
?>
```

+ Inheritance and Interface

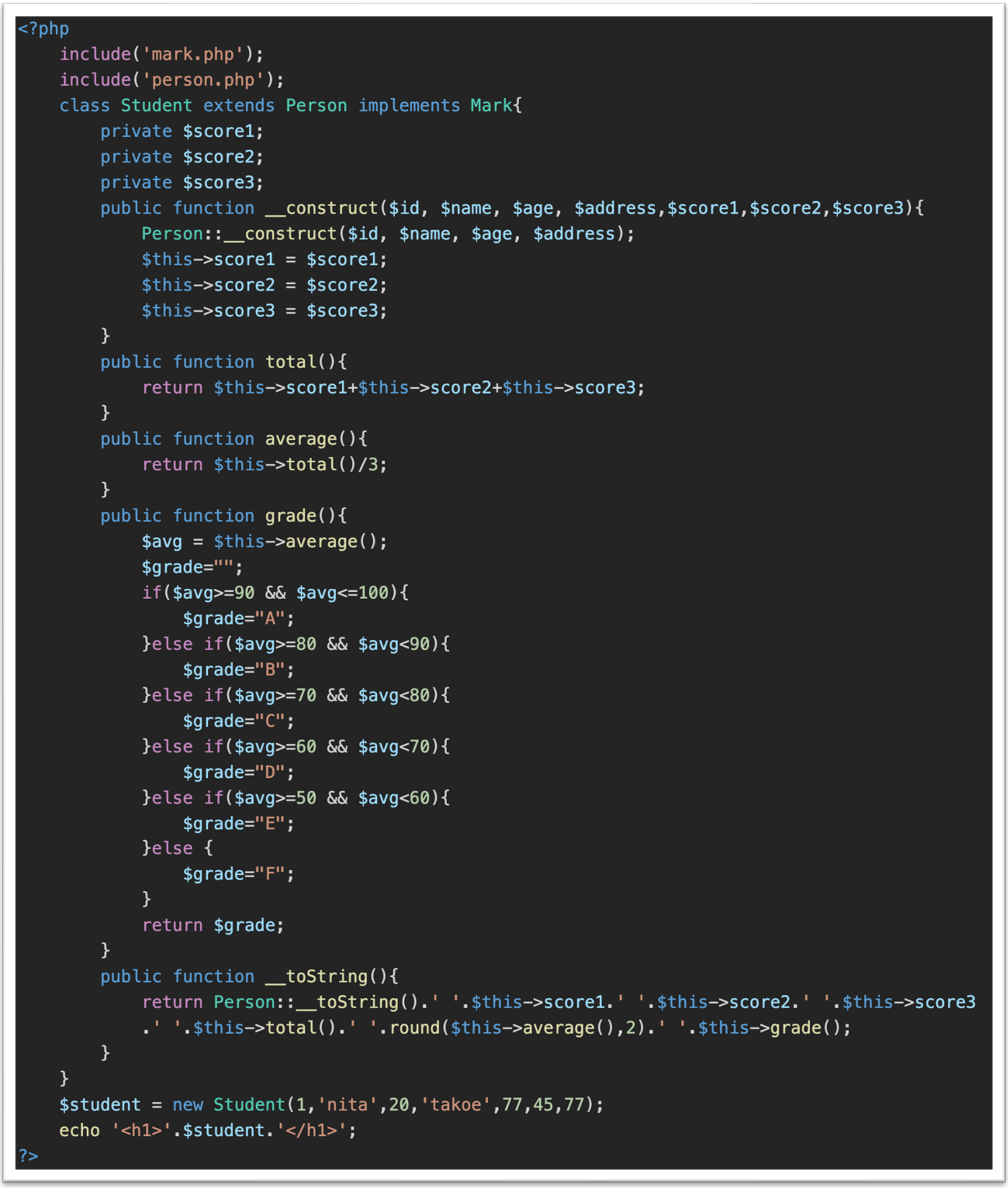


```php
<?php
    class Person{
        protected $id;
        protected $name;
        protected $age;
        protected $address;
        public function __construct($id,$name,$age,$address){
            $this->id      = $id;
            $this->name    = $name;
            $this->age     = $age;
            $this->address = $address;
        }
        public function setAge($age){
            if($age>100){
                echo '<p>Something wrong</p>';
            }else{
                $this->age = $age;
            }
        }
        public function getAddress(){
            return $this->address;
        }
        public function __toString(){
            return $this->id.' '.$this->name.' '.$this->age.' '.$this->address;
        }
    }
?>
```

- **interface** and **abstract** use for set rule to class
- **include(file_name)** use for get file

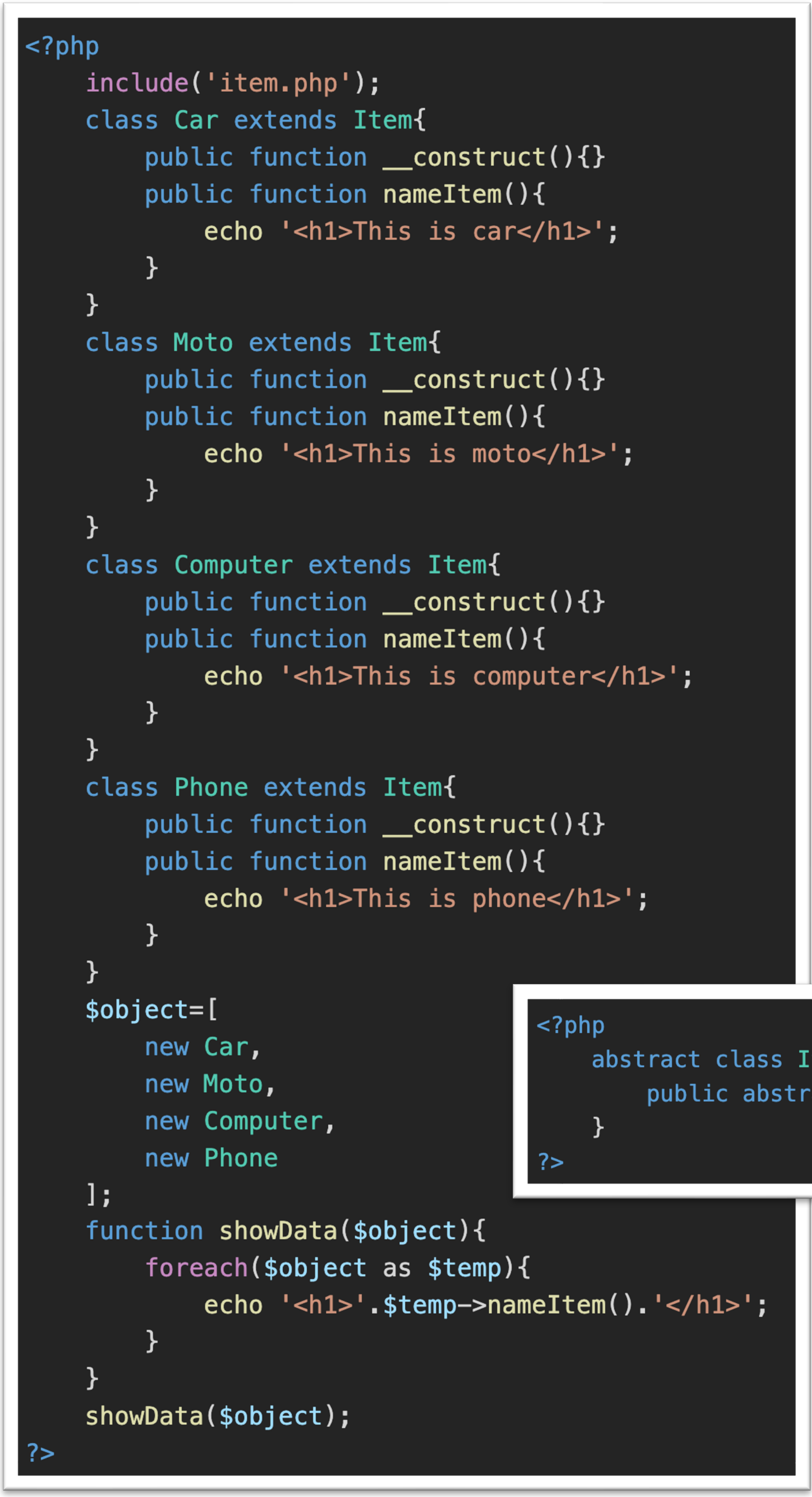


```php
<?php
    interface Mark{
        public function total();
        public function average();
        public function grade();
    }
?>
```

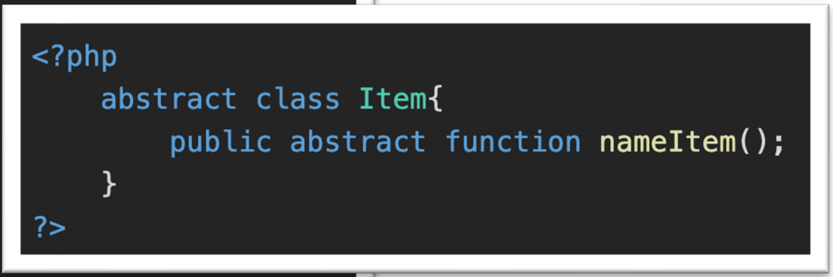


```php
<?php
    include('mark.php');
    include('person.php');
    class Student extends Person implements Mark{
        private $score1;
        private $score2;
        private $score3;
        public function __construct($id, $name, $age, $address,$score1,$score2,$score3){
            Person::__construct($id, $name, $age, $address);
            $this->score1 = $score1;
            $this->score2 = $score2;
            $this->score3 = $score3;
        }
        public function total(){
            return $this->score1+$this->score2+$this->score3;
        }
        public function average(){
            return $this->total()/3;
        }
        public function grade(){
            $avg = $this->average();
            $grade="";
            if($avg>=90 && $avg<=100){
                $grade="A";
            }else if($avg>=80 && $avg<90){
                $grade="B";
            }else if($avg>=70 && $avg<80){
                $grade="C";
            }else if($avg>=60 && $avg<70){
                $grade="D";
            }else if($avg>=50 && $avg<60){
                $grade="E";
            }else {
                $grade="F";
            }
            return $grade;
        }
        public function __toString(){
            return Person::__toString().' '.$this->score1.' '.$this->score2.' '.$this->score3
            .' '.$this->total().' '.round($this->average(),2).' '.$this->grade();
        }
    }
    $student = new Student(1,'nita',20,'takoe',77,45,77);
    echo '<h1>'.$student.'</h1>';
?>
```

+ Polymorphism with Abstract



```php
<?php
    include('item.php');
    class Car extends Item{
        public function __construct(){}
        public function nameItem(){
            echo '<h1>This is car</h1>';
        }
    }
    class Moto extends Item{
        public function __construct(){}
        public function nameItem(){
            echo '<h1>This is moto</h1>';
        }
    }
    class Computer extends Item{
        public function __construct(){}
        public function nameItem(){
            echo '<h1>This is computer</h1>';
        }
    }
    class Phone extends Item{
        public function __construct(){}
        public function nameItem(){
            echo '<h1>This is phone</h1>';
        }
    }
    $object=[
        new Car,
        new Moto,
        new Computer,
        new Phone
    ];
    function showData($object){
        foreach($object as $temp){
            echo '<h1>'.$temp->nameItem().'</h1>';
        }
    }
    showData($object);
?>
```



```php
<?php
    abstract class Item{
        public abstract function nameItem();
    }
?>
```