بسم الله الرحمن الرحيم

اتكلمنا عن ال polymorphism ك مفهوم وعرفنا انه عباره عن نفس ال method لكنها ف كل class بتعمل وظيفه مختلفه عن ال class التاني ,, كمان عرفنا ال abstraction وانه عباره عن تعريف ال methods من غير ما نعرف ال implementation بتاعها .

تعالي نتكلم بقي عن ازاي الكلام دا ممكن نطبقه بشكل واضح كدا ,, وهنعرف ايه هو ال interface وال abstract class وازاي ممكن يساعدونا ف project بتاعنا .
لكن الاول عايزك تتخيل معايا اننا e-commerce وعندنا موردين , لكن ال data اللي احنا بنحتاجها من كل مورد مختلفه عن التاني ,, يعني مثلا واحد عايزين نخزن رقم موبيله بس والتاني الايميل بتاعه والفاكس كمان ,, كمان ال methods اللي كل واحد فيهم بيعملها مختلفه عن التاني ف طريقه ال implement بتاعتها وكمان عددهم مختلف .
كل دا بيخلينا نفكر ازاي ممكن نعمل blueprint لل class ,, ايه دا انت مش قولت ان ال class دا هو ال blueprint لل object ؟ ايوه صح ,, احنا عايزين نعمل blueprint لل blueprint 😀 ونخلي فيه عندنا زي standard كدا احنا ماشيين عليه ,, يعني عايزين نعمل blueprint لل e-commerce بتاع المورد دا ونخلي كل الموردين عندهم method بتعمل print و method تانيه بتعمل getAllData .. عشان لمه نيجي نستخدمها ف ال code بتاعنا احنا عايزين نقول getAllData مباشره من غير ما اعرف انا ف انهي مرود ولا ال implementation بتاعها ايه .

هنا عندنا 2 options وهم :

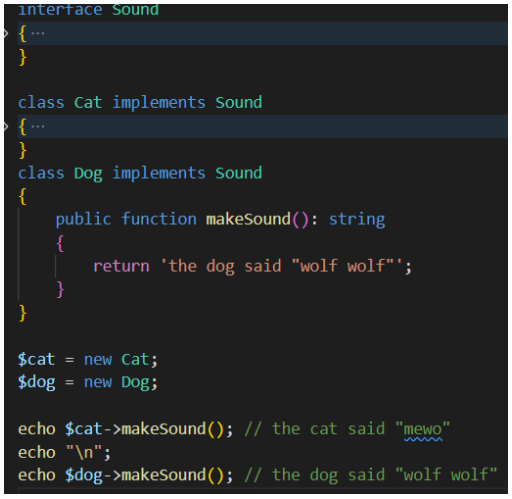
- ال interface .
- وال abstract class .

تعالي نتكلم بقي عن ال interface :

ال interface هو حاجه بتسمحلي احدد ايه هي ال methods اللي المفروض ال class اللي هيعمل implementation لل interface دا انه يعملها (وهنا الموضوع اجباري ان لو ال interface دا فيه print method and getAllData method ف ال class اللي بيعمل implementation لل interface دا – لازم – يعمل implement لل 2 methods دول) .

وهنا الخصائص بتاعه ال interface هتكون ك التالي :

- لازم كل ال methods تكون public access modifier .
- كل ال methods اللي ف ال interface هتكون بطبيعتها abstract يعني ملهاش implementation .
- ال interface ملهوش property .
- لكن نقدر نحدد ال datatype بتاعه ال methods parameter and the return type of the method .
- وزي ما قولنا ان ال class بيعمل implement للـ interface مش بيعمل extend منه .

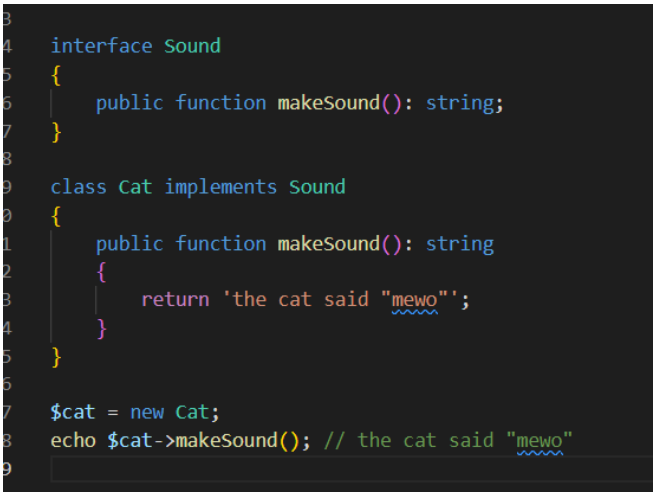


```php
interface Sound
{...
}

class Cat implements Sound
{...
}
class Dog implements Sound
{
    public function makeSound(): string
    {
        return 'the dog said "wolf wolf"';
    }
}

$cat = new Cat;
$dog = new Dog;

echo $cat->makeSound(); // the cat said "mewo"
echo "\n";
echo $dog->makeSound(); // the dog said "wolf wolf"
```



```php

interface Sound
{
    public function makeSound(): string;
}

class Cat implements Sound
{
    public function makeSound(): string
    {
        return 'the cat said "mewo"';
    }
}

$cat = new Cat;
echo $cat->makeSound(); // the cat said "mewo"

```

ف المثال دا انا عندي interface Sound فيه makeSound method وبنعملها implement ب Cat class and Dog class – وهنا بيتحقق مفهوم ال abstract وهو اني عندي method لكن معرفش ال implementation بتاعها ايه – وكل class فيهم بيعمل implement مختلف تماما عن التاني – وهنا بيتحقق مفهوم ال polymorphism واني عندي implementation مختلف لنفس ال method – .

تعالي بقي نتكلم عن ال abstract class :

ال abstract class هو class عندي (لكن منقدرش ناخد منه object) بنعمل inherent منه وبيكون جواه ع الاقل method واحده abstract – عرفنا ان ال abstract method هي ال method اللي بيكون ملهاش implementation – وال classes اللي بتورث من ال abstract class دا – .

وخصائص ال abstract class :

- ممكن نحط فيه properties وال class اللي هيعمل extend هيورثهم مفيش مشاكل.
- ممكن نستخدم ال public وال protected لكن لمه نيجي نعمل implement لازم تاخد بالك من :
    - لل child class لازم يكون عنده نفس ال access modifier لو اقل .

      طب يعني ايه اقل ؟ يعني بالترتيب كدا من الاعلي للاقل private > protected > public

      ال private هي اعلي حاجه وال public هي اقل حاجه

      ال access modifier لازم يكون نفس المستوي او اقل منه .
- ممكن نعمل implement لل methods لو احنا عايزين وال child يقدر يورثهم ولو حب يعدل عليهم مش هيقول لا .
- ال abstract method بنكتب قبلها abstract ومبنعملهاش implementation ونقدر نحدد فيها ال return type and parameters types برضو زي اي class عادي .

```php
abstract class Vehicle
{
    protected $make;
    protected $model;

    public function __construct($make, $model)
    {
        $this->make = $make;
        $this->model = $model;
    }

    public function getMake()
    {
        return $this->make;
    }

    public function getModel()
    {
        return $this->model;
    }

    public function stop()
    {
        return "The {$this->make} {$this->model} stopped successfully";
    }

    abstract public function start();
}
```

```php
class Car extends Vehicle
{
    public function start()
    {
        return "The {$this->make} {$this->model} started successfully";
    }
}

$car = new Car('BMW', '2025');

echo $car->start(); // The BMW 2025 started successfully
echo "\n";
echo $car->stop(); // The BMW 2025 stopped successfully
```

- ف المثال دا احنا عندنا ال abstract class واللي فيه properties ال Car قدرت انها تورثها .
- كمان فيه predefined method موجوده ف ال abstract class قدرنا نورثها برضو .
- وفيه عندي abstract method ملهاش implementation لمه جينا نعمل extend ليها ف ال Car class عملنلها implementation .

<u>تعالي بقي ملخص سريع كدا :</u>

ال interface وال abstract class عبارة عن blueprint لل class لكن ببعض الاختلافات زي :

- ال abstract class ممكن نعمل فيه implement ل methods فيه وكمان ممكن نحط فيه properties .. ع عكس ال interface منقدرش نعمل implement لل method ولا نضيف properties .
- نقدر نعمل implementation لاكتر من interface ف نفس الوقت ,, لكن منقدرش نعمل غير extend مره واحده بس ل abstract class واحد فقط .
- ف ال abstract class نقدر نستخدم ال protected and public access modifiers لكن منقدرش نسخدم ال private وف ال interface هو public بس .
- نقدر نستخدم ال abstract method لمه يكون عندنا ال implementation بتاعها مشترك , ف حين ال interface بنستخدمها لمه يكون عندنا methods مشتركه لكن ال implementation مختلفه من class للتاني .