SASF
Sudamericana de Software S.A.

Capacitación PHP/SYMFONY

Módulo: POO

Programación Orientada a Objetos

- Objetos que contienen data y funciones
- Fácil y rápido de ejecutar
- Estructura clara de programas
- D.R.Y
- Aplicaciones reutilizables

Pilares

- **Abstracción:** La abstracción oculta al usuario la funcionalidad interna de una aplicación.
- **Herencia:** Definir múltiples subclases a partir de una clase ya definida.
- **Polimorfismo:** Modificar ligeramente los métodos y atributos de las subclases previamente definidas en la superclase
- **Encapsulamiento:** La encapsulación es el proceso en el que protegemos la integridad interna de los datos en una clase

Clases y Objetos

- Una clase es una plantilla de un objeto
- Cuando se crea un objeto, hereda las propiedades y comportamientos de la clase.
- Cada objeto tendrá diferentes valores.

Constructor

- Permite inicializar las propiedades de un objeto al crear el objeto

```
__construct()
```


- Es una función, PHP la llama automáticamente cuando se crea un objeto.

SASF
Sudamericana de Software S.A.

Ejemplo Clase

```
class Persona {
public $nombre;
public $edad;
  public function___construct($nombre, $edad) {
    $this->nombre = $nombre;
    $this->edad = $edad;
  }
  public function presentacion() {
    return “Mi nombre es: " . $this-> . “ y tengo " . $this->edad . “
     años!";
  }
}
```

Ejemplo Objeto

```php
$personaMilen = new Persona("Milen", "23");
echo $personaMilen -> presentacion();
echo "<br>";
$personaJuan = new Persona("Juan", "30");
echo $personaJuan -> presentacion();
```

Modificadores de Acceso

- **Public:** Se puede acceder a la propiedad o al método desde cualquier lugar. Default.
- **Protected:** Se puede acceder a la propiedad o método dentro de la clase y por clases derivadas de esa clase
- **Private:** Solo se puede acceder a la propiedad o método dentro de la clase

Modificadores de Acceso

- **Public:** Se puede acceder a la propiedad o al método desde cualquier lugar. Default.
- **Protected:** Se puede acceder a la propiedad o método dentro de la clase y por clases derivadas de esa clase
- **Private:** Solo se puede acceder a la propiedad o método dentro de la clase

SASF
Sudamericana de Software S.A.

Ejemplo Encapsulamiento

```php
class Persona {
    private $nombre;
    private $edad;

    public function setNombre($nombre){
        $this->nombre = $nombre;
    }
    public function getNombre(){
        return $this->nombre;
    }

    public function setEdad($edad){
        $this->edad = $edad;
    }
    public function getEdad(){
        return $this->edad;
    }
}
```

Herencia

- Clases derivadas de otras clases
- Clase hija hereda propiedades y métodos con acceso public y protected
- Se usa por medio de keyword ***extends.***

Clases y Métodos Abstractos

- Son clases padres que tienen métodos que están vacíos.
- Debe contener mínimo 1 método abstracto
- No pueden ser instanciadas
- Keyword: ***abstract***

Reglas de Clases Abstractas

- El método de la clase hija debe definirse con el mismo nombre y re declarar el método abstracto de la clase padre
- El método de la clase hija debe definirse con el mismo modificador de acceso o uno menos restringido.
- El número de argumentos requeridos debe ser el mismo. Sin embargo, la clase secundaria puede tener argumentos opcionales además

SASF
Sudamericana de Software S.A.

Ejemplo Herencia y Clase Abstracta

```php
abstract class FiguraGeometrica{
    private $area;
    private $perimetro;

    abstract public function calculaArea();
    abstract public function calculaPerimetro();
}
```

```php
class Cuadrado extends FiguraGeometrica{
    private $lado = 4;

    public function calculaArea(){
        $this->area = $this->lado * $this->lado;
    }

    public function calculaPerimetro(){
        $this->perimetro = $this->lado * 4;
    }
}
```

```php
class Triangulo extends FiguraGeometrica{
    private $base = 3;
    private $altura = 5;
    private $ladoA = 7;
    private $ladoB = 9;
    private $ladoC = 10;

    public function calculaArea(){
        $this->area = $this->ladoA + $this->ladoB + $this->ladoC;
    }

    public function calculaPerimetro(){
        $this->perimetro = ($this->base * $this->altura) / 2;
    }
}
```

SASF
Sudamericana de Software S.A.

Ejemplo Herencia y Clase Abstracta

```php
$cuadrado = new Cuadrado();
$cuadrado->calculaArea();
$cuadrado->calculaPerimetro();
echo "Area del cuadrado: " . $cuadrado->area;
echo "<br>";
echo "Perimetro del cuadrado: " . $cuadrado->perimetro;

echo "<br><br>";

$triangulo = new Triangulo();
$triangulo->calculaArea();
$triangulo->calculaPerimetro();
echo "Area del triangulo: " . $triangulo->area;
echo "<br>";
echo "Perimetro del triangulo: " . $triangulo->perimetro;
```

Area del cuadrado: 16
Perimetro del cuadrado: 16

Area del triangulo: 26
Perimetro del triangulo: 7.5

Interfaces

- Permiten especificar qué métodos debe implementar una clase
- Las interfaces facilitan el uso de una variedad de clases diferentes de una misma manera.
- Keyword declaración: ***interface***
- Keyword para llamado: ***implements***

Interface Vs. Abstract Class

- Interfaces no tienen propiedades
- Todos los métodos de la interface son públicos
- Las clases pueden implementar una interfaz mientras heredan de otra clase al mismo tiempo

SASF
Sudamericana de Software S.A.

Ejemplo Interface

```php
interface EnvioNotificaciones{
    public function obtenerConfiguracion();
    public function enviaNotificacion();
}
```

```php
class NotificacionSMS implements
EnvioNotificaciones{
    public function obtenerConfiguracion(){
        echo "Se obtiene API proveedor para envio
de notificaciones por SMS";
    }

    public function enviaNotificacion(){
        echo "Se envia notificación por SMS";
    }
}
```

```php
class NotificacionCorreo implements EnvioNotificaciones{
    public function obtenerConfiguracion(){
        echo "Se obtiene servidor para envio de notificaciones
por Correo";
    }

    public function enviaNotificacion(){
        echo "Se envia notificación por correo";
    }
}
```

```php
class NotificacionWhatsapp implements EnvioNotificaciones{
    public function obtenerConfiguracion(){
        echo "Se obtiene API para envio de notificaciones por
Whatsapp";
    }

    public function enviaNotificacion(){
        echo "Se envia notificación por Whatsapp";
    }
}
```

Ejemplo Interface

```php
$notificacionSMS = new NotificacionSMS();
$notificacionSMS->obtenerConfiguracion();
echo "<br>";
$notificacionSMS->enviaNotificacion();

echo "<br><br>";

$notificacionCorreo = new NotificacionCorreo();
$notificacionCorreo->obtenerConfiguracion();
echo "<br>";
$notificacionCorreo->enviaNotificacion();

echo "<br><br>";

$notificacionWs = new NotificacionWhatsapp();
$notificacionWs->obtenerConfiguracion();
echo "<br>";
$notificacionWs->enviaNotificacion();
```

```
Se obtiene API proveedor para envio de notificaciones por SMS
Se envia notificación por SMS

Se obtiene servidor para envio de notificaciones por Correo
Se envia notificación por correo

Se obtiene API para envio de notificaciones por Whatsapp
Se envia notificación por Whatsapp
```

Delegación

- Llamamos delegación a la situación en la que una clase contiene (como atributos) una o más instancias de otra clase, a las que *delegará* parte de sus funcionalidades.

SASF
Sudamericana de Software S.A.

Ejemplo Delegación

```php
class NotificacionProvider{
    public function consultaConfiguracion($proveedor){
        return "Se obtiene API proveedor para envio de notificaciones por SMS del proveedor " . $proveedor;
    }
}

class NotificacionSMS implements EnvioNotificaciones{
    private $notificacionProv;

    public function obtenerConfiguracion(){
        $this->notificacionProv = new NotificacionProvider();
        echo  $this->notificacionProv->consultaConfiguracion("CLARO");
    }
    public function enviaNotificacion(){
        echo "Se envia notificación por SMS";
    }
}
```

```
Se obtiene API proveedor para envio de notificaciones por SMS
Se envia notificación por SMS

Se obtiene servidor para envio de notificaciones por Correo
Se envia notificación por correo
```

Ejemplo Delegación

```php
$notificacionSMS = new NotificacionSMS();
$notificacionSMS->obtenerConfiguracion();
echo "<br>";
$notificacionSMS->enviaNotificacion();
```

Se obtiene API proveedor para envio de notificaciones por SMS del proveedor CLARO
Se envia notificación por SMS