POO/OOP

Programación Orientado a Objetos

Object-Oriented Programming

Vocabulario

- Clase, objeto
- Ejemplar de clase, instancia de clase, ejemplarizar una clase, instanciar una clase
- Modularización
- Encapsulamiento / encapsulación
- Herencia
- Poliformismo

Encapsulamiento

- El concepto de **Encapsulamiento** está relacionado con otro concepto: el de **visibilidad**
- Como aproximación debemos saber que los atributos y métodos por defecto son públicos. Es decir se puede acceder a ellos desde fuera de la clase.
- En la **POO** esto no es lo deseable. Lo ideal es que las variables de campo o atributos se encuentren protegidos, esto es, que sean accesibles desde dentro de la clase, pero no desde fuera. Posibilidades de los **modificadores de acceso**:
  - **public**
    + Por defecto todas las propiedades y funciones son públicos
  - **Private**
    - Accesibles desde dentro de la clase
  - **protected**
    # Accesible desde la propia clase o un descendiente de éste.

Manipulación de atributos

Si los campos son privados. ¿Cómo los podeos manipular?

Método constructor

Como vimos anteriormente el **método constructor** sirve para establecer un **estado inicial** de la **instancia de la clase**.

Pero como función/método que es nos permite el paso de parámetros con el fin de establecer el valor de las propiedades al objeto.

```php
public function __construct(string $paramNombre = "Sin nombre",float $paramSueldo = 0.0, int $ano = 1, int $mes = 1, int $dia = 1){
   $this->nombre = $paramNombre;
   $this->sueldo = $paramSueldo;
   ////DateTime::createFromFormat //método estático --> ahora da igual, ya se verá
   $this->altaContrato = DateTime::createFromFormat('Y-m-d',"$ano-$mes-$dia");

}
```

Manipulación de atributos

Setters y Getters

Otra opción es crear **métodos** o **funciones de clase** que nos permita manipular susodichos ATRIBUTOS/CAMPOS/PROPIEDADES.

Getters

- Función: **devolver** el valor de las propiedades de los objetos
- Sintaxis: public function nombre_método():tipo_dato_a_devolver { código + return}

```
/**
     * GETTERS o captadores
     */
    /**
     * Captador del nombre
     *
     * @return string
     */
    public function getNombre():string{
        return $this->nombre;
    }
```

Setters

- Función: **Modifica el valor** de las propiedades de los objetos
- Sintaxis: public function nombre_método():void{ código}
  - ¿Qué indica void? Indica que el método no devuelve ningún valor
  - Al no devolver valor no la lleva palabra reservada **return**

```
/**
* SETTERS o definidores
* Un definidor por norma nunca retorna ningún valor
* Tampoco es necesario ponerse a construir setters a lo loco. Por ejemplo:
* El sueldo de un empleado puede cambiar a lo largo del tiempo
* No así la fecha de alta o el nombre
*/
    public function setSueldo(float $s):void{
        $this->sueldo = $s;
    }
```

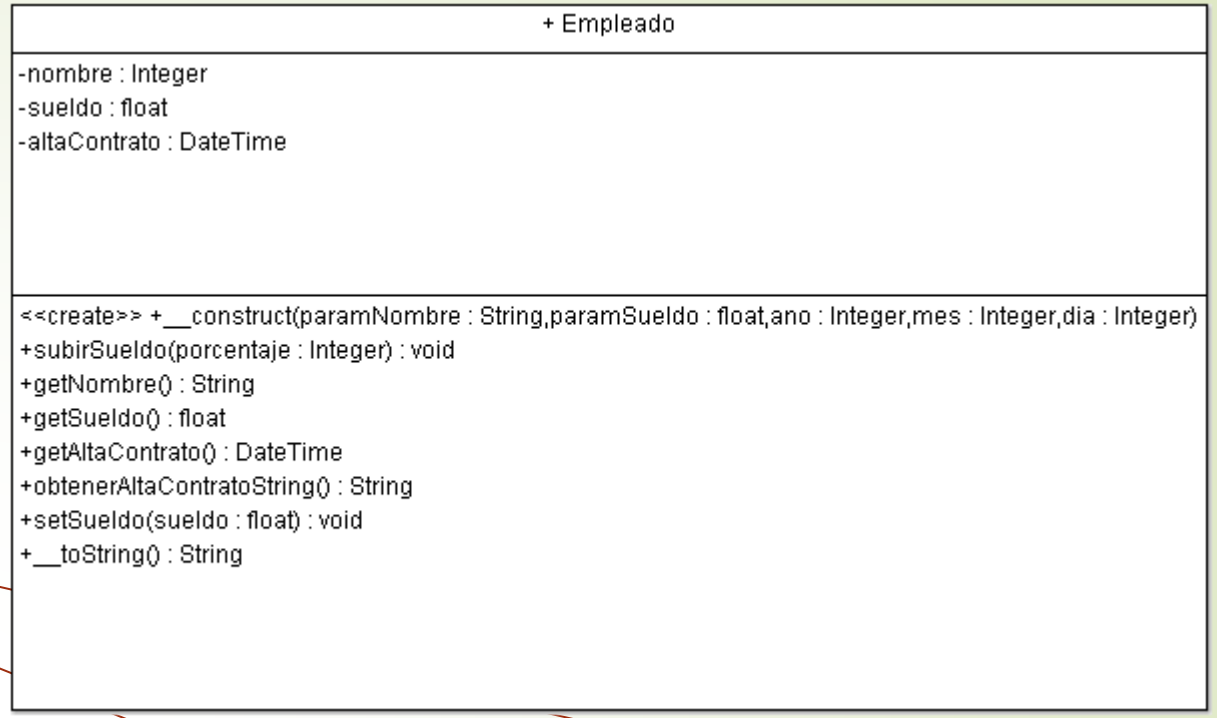
function __toString(void):string

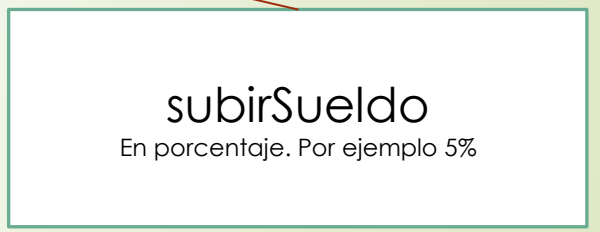
- Este **método** permite convertir **un objeto a una cadena**
- Invocado cada vez que el objeto es utilizado como una cadena de texto.
  Por ejemplo, utilizando la función echo:

```
echo $e1; //$e1 es una instancia de la clase Empleado
```

```
/**
 * Método automático __toString
 */
public function __toString():string{
    return "<ul>
                <li>Nombre: <em>$this->nombre</em></li>
                <li>Sueldo: <em>$this->sueldo €</em></li>
                <li>Alta: <em>{$this->obtenerAltaContratoString()}</em></li>
            </ul>";
}
```

Clase Empleado





En ArgoUml se puden mostrar los modificadores de visibilidad:
+ indica atributo o método público
- Indica atributo o método privado
Indica atributo o método protegido

Crear el UML y el script de la clase Libro

- Atributos de la clase Libro (todos ellos privados)
  - ISBN
  - Título
  - Autor
  - Número de páginas
  - Fecha de alta (coincidirá con la des sistema)
- Métodos necesarios
  - Contructor
  - Getters y setter considerados necesarios
  - Método __**toString** que deberá mostrar un texto como el siguiente
    "El Libro **Don Quijote de la Mancha** con ISBN **324343** escrito por **Miguel de Cervantes** tiene **788** páginas"
- Crear mínimo dos instancias de la clase e indicar cuál de ellas tiene el mayor número de páginas