

EJERCICIOS POO EN PHP

EJERCICIO 1 (POO, herencia, elementos estáticos):

Realiza el código correspondiente al siguiente diagrama de clases:



Notas:

- La clase empleado hereda de la clase persona
- En la clase empleado tenemos una variable estática `suelo_impuestos` (float), cuyo valor por defecto es 1200. Tendrá un setter estático para la misma. *Nota:* En UML se indica que son estáticos mediante subrayado.
- Los atributos `nombre` y `apellidos` son protegidos (en UML se indica con `#`)
- El atributo `teléfonos` es un array de strings, y el atributo `salario` es float

Métodos:

- `Get_nombreCompleto`: Retornará un string con la concatenación de `nombre` y `apellidos`, separados por un espacio en blanco.
- `Salario` es privado, debe manipularse mediante su setter y su getter
- `Paga_impuestos`: Retorna true si `salario` es mayor que `suelo_impuestos`. En caso contrario retorna false.
- `Ania_de_telefono`: Añade un teléfono al array `telefonos`.

- Lista_telefonos: Retorna un string con todos los teléfonos separados por espacios
- Borra_telefonos: Borra todos los teléfonos de la lista telefonos
- Set_sueldo_impuestos: Permite modificar la variable estática privada.
- Informe_HTML: Retorna un string cuyo contenido es la información del empleado, en HTML, para visualizar como se muestra en la siguiente figura:

marta olmedilla

Teléfonos:

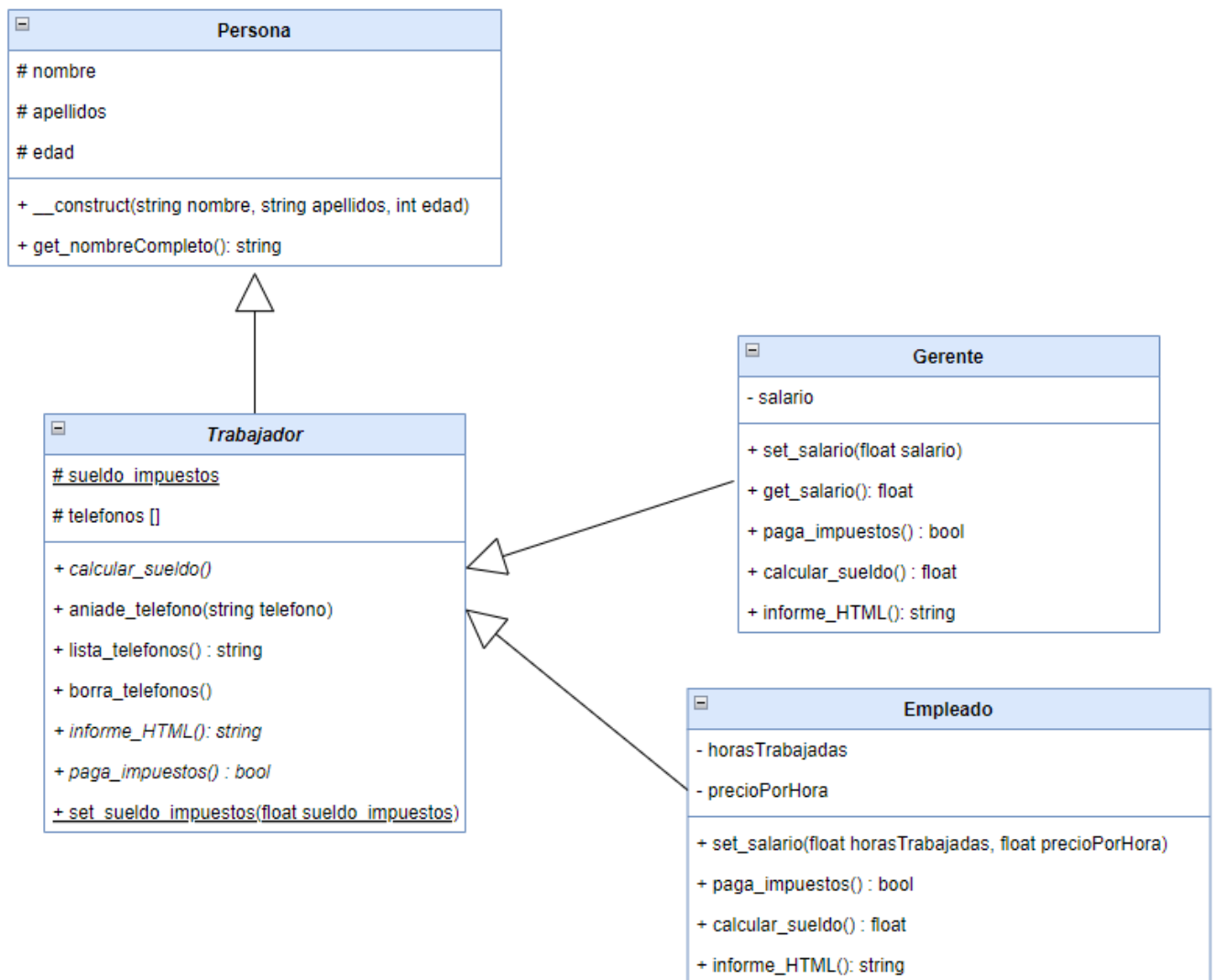
- 91 568 65 69 65
- 698 89 55 66

Salario: 2000

marta olmedilla

EJERCICIO 2 (Clases abstractas):

Realiza el código correspondiente al siguiente diagrama de clases:



Recordatorio de UML:

- Se utiliza la cursiva para indicar que el elemento es abstracto. Así, **Trabajador** es una clase abstracta, con dos métodos abstractos, y el resto no abstractos.
- Se utiliza el subrayado para indicar que un elemento es estático. Así, tenemos un atributo estático privado y su setter estático.
- El símbolo “#” significa protegido, el “-” significa privado, y el “+” significa público

Descripción de métodos:

En **Persona**:

- El `get_nombreCompleto` retorna nombre y apellidos separados por un espacio

En **Trabajador**:

- El método `aniade_telefono` añade un teléfono al array “telefonos”

- El método `lista_telefonos` retorna un string con los teléfonos separados por espacios
- El método `borra_telefonos` elimina todos los telefonos, dejando un array vacío
- Los métodos `calcular_sueldo`, `informe_HTML` y `paga_impuestos` son abstractos
- El método estático `set_sueldo_impuestos` modifica la variable estática `sueldo_impuestos` con el valor pasado por parámetro

En Gerente:

- `salario` es un atributo privado con su setter y su getter. Por defecto tendrá el valor de 3000
- El método `calcular_sueldo` retorna el sueldo, calculado como:
 - o $\text{Salario} + \text{Salario} * \text{edad} / 100$
- El método `paga_impuestos` retorna true si el sueldo es mayor que el *sueldo_impuestos* de la clase `Persona`.

En Empleado:

- `HorasTrabajadas` y `precioPorHora` son atributos privados. Por defecto tendrán los valores: 160 horas y 10 euros por hora.
- Se setean mediante `set_salario`, que recibirá los valores de ambos por parámetro
- El método `calcular_sueldo` retorna el sueldo, calculado como:
 - o $\text{horasTrabajadas} * \text{precioPorHora}$
- El método `paga_impuestos` retorna true si el sueldo es mayor que el *sueldo_impuestos* de la clase `Persona`

Los métodos `informe_HTML` se describen en el siguiente apartado.

Probando las clases

Primer paso:

Instanciamos un objeto de la clase `Gerente`, y otro de la clase `Empleado`.

Luego utilizamos el `informe_HTML` de cada uno, y el resultado debe ser el siguiente:

GERENTE: Pedro Gómez González

Teléfonos:

Edad:43

Salario Base: 3000

Sueldo: 4290

Paga Impuestos: SI

EMPLEADO: Marta Sánchez

Teléfonos:

Edad:56

Horas trabajadas: 160

Precio por hora: 10

Sueldo: 1600

Paga Impuestos: SI

Segundo paso:

Utilizamos los métodos necesarios para:

- Modificar el salario mínimo de pago de impuestos a 3000
- Modificar el salario del gerente a 4500
- Modificar el importe por hora del empleado a 8 euros
- Añadir varios teléfonos al gerente y al empleado

Finalmente realizamos un informe_HTML de ambos, el resultado debe ser similar a:

GERENTE: Pedro Gómez González

Teléfonos:

- 659443945
- 3324422934
- 91 234234223

Edad:43

Salario Base: 4500

Sueldo: 6435

Paga Impuestos: SI

EMPLEADO: Marta Sánchez

Teléfonos:

- 91338923432
- 632234523033

Edad:56

Horas trabajadas: 160

Precio por hora: 8

Sueldo: 1280

Paga Impuestos: NO

EJERCICIO 3: Sobreescritura de constructor

Vas a coger el diagrama de clases del ejercicio anterior (el 2), y vas a modificarlo de forma que tanto la clase Empleado como la clase Gerente tengan un constructor para inicializar sus atributos privados.

Luego modifica el código y comprueba que funciona.