



AVISO: Las tareas son trabajo individual y no deben ser compartidas con otros compañer@s. Si se detecta o se sospecha que una tarea es copia total o parcial de otra, serán anuladas las 2 tareas, independientemente de quién realizara la versión original. La nota de ambas tareas será de 0 puntos

En los apartados en los que es necesario entregar las capturas de pantalla, éstas deben tener como fondo de pantalla la plataforma de fpadistancia con tu usuario mostrando claramente la foto de tu perfil. Aquellos apartados/subapartados que no cumplan esta condición no serán corregidos. Por ejemplo:

CONSEJO PARA NOVAT@S: RECUERDA IR GUARDANDO TU TRABAJO (HACIENDO CLIC EN EL DISKETE DE LA BARRA DE HERRAMIENTAS) CADA POCO TIEMPO PARA NO PERDERLO.

Recuerda subir también el código fuente, aparte de GitHub, con este enunciado de la práctica dentro de un fichero .zip.

Repo GitHub:

https://github.com/diegorodbar/Trabajos-DWCS/tree/main/ud3-ejercicio1

PREGUNTA 1: (6,00 puntos)

Deseamos crear una aplicación que modelará o guardará información sobre las personas que conforman la comunidad educativa de un instituto de enseñanza secundaria. En concreto, tenemos los siguientes tipos de personas, de los que queremos guardar la información:

para todas las personas:

- nombre, apellido1 y apellido2
- fecha nacimiento
- DNI o documento de identificación
- Dirección
- Teléfono(s)
- sexo

administrativos:

años de servicio

conserjes:

- años de servicio
- personal de limpieza:
- años de servicio

profesorado:

- años de servicio
- materias que imparte
- cargo directivo: ninguno, dirección, secretariado, jefatura estudios diurno, jefatura estudios personas adultas, vicedirección

alumnado de la ESO:

- curso
- grupo

alumnado de Bachillerato

- · curso
- arupo

alumnado de Formación Profesional

- ciclo formativo
- curso
- grupo

Crea el código de una jerarquía de clases que permite modelar la información anterior, diseñando la/s clase/s abstracta/s, métodos setter/getter, propiedades/atributos, métodos auxiliares, etcétera necesarios. Diseña tu jerarquía de clases teniendo en cuenta que se pueda usar polimorfismo.

Ten además en cuenta las cuestiones siguientes:

todas las clases que modelan lo anterior, tendrán un método/función generarAlAzar() que devolverá un





objeto de esa clase con información inventada (pero coherente, es decir, con formato que respete la semántica del elemento a representar).

todas las clases implementarán un método numeroObjetosCreado () que devuelva la cantidad de objetos que se han creado de esa clase en concreto.

implementa el método __toString() para cada clase.

todas las clases que modelan lo anterior, tendrán un método trabajar() que devolverá un string o cadena indicando lo que hacer una persona de esa clase (por ejemplo: "Soy un estudiante de la ESO y estoy estudiando" ó "Soy una estudiante de la ESO y estoy estudiando"; es decir, se tendrá en cuenta la propiedad "sexo" para imprimir un mensaje correcto).

Documenta todo el código.

Observaciones:

A continuación, se presenta una explicación de la estructura y funcionamiento de un programa en PHP que modela diferentes tipos de personas en un entorno académico, utilizando clases y métodos para gestionar atributos y operaciones específicas de cada tipo de persona. Este programa se organiza en una jerarquía de clases para crear objetos de manera eficiente, asegurando que cada clase contenga tanto atributos como métodos específicos y que se respete la herencia y polimorfismo en la estructura de clases.

Clase Abstracta Persona

La clase Persona es una clase abstracta que sirve como base para todas las demás. En este contexto, una clase abstracta define un modelo genérico que no se debe instanciar directamente, sino que se usa como referencia para otras clases específicas. La clase Persona representa los atributos y comportamientos básicos comunes a todas las personas en un entorno académico.

Los atributos de esta clase incluyen:

nombre, apellido1, apellido2: Para almacenar el nombre y los dos apellidos de cada persona.

fechaNacimiento: Guarda la fecha de nacimiento de la persona.

dni: Número de identificación. direccion: Dirección física.

telefonos: Un array que almacena números telefónicos.

sexo: Define el género de la persona.

Además, Persona tiene un atributo estático \$contador que lleva el registro del número de objetos creados de esta clase y sus subclases.

Constructor de Persona

El constructor recibe todos los atributos como parámetros, los asigna a la instancia, e incrementa el contador de objetos creados. Este contador es estático, por lo que se comparte entre todas las instancias de Persona y sus clases derivadas.

Métodos de Persona

Getters y Setters: Son métodos para acceder y modificar los atributos privados de la clase. Siguen el patrón de encapsulamiento, que permite interactuar con los atributos de manera controlada.

generarAlAzar: Método estático que crea un objeto Persona con datos aleatorios. Este método utiliza listas de nombres, apellidos, y una fecha generada al azar para simular datos realistas.

numeroObjetosCreado: Método estático que devuelve el valor del contador, informando sobre la cantidad de objetos creados.

__toString: Método mágico que retorna una representación en cadena de la información de la persona. trabajar: Método que simula una acción genérica de trabajar, devolviendo un mensaje básico.





Clases Derivadas

Cada clase derivada representa un tipo específico de persona en el entorno académico. Todas extienden la clase Persona, heredando así sus atributos y métodos básicos, pero también agregando características y comportamientos específicos.

Clase Administrativo

Atributo adicional: aniosServicio, que guarda los años de servicio.

Método trabajar: Implementado para devolver un mensaje específico según el sexo de la persona.

generarAlAzar: Este método extiende la funcionalidad de Persona::generarAlAzar para crear un objeto Administrativo con un valor aleatorio de años de servicio.

toString: Método que representa los datos del objeto como una cadena, incluyendo los años de servicio.

Clase Conserje

La clase Conserje es similar a Administrativo, pero su función trabajar devuelve un mensaje que describe sus tareas específicas de conserjería. Los métodos y atributos adicionales son equivalentes a los de la clase Administrativo.

Clase PersonalLimpieza

Esta clase representa al personal de limpieza y sigue el mismo patrón de herencia que Administrativo y Conserje. La función trabajar devuelve un mensaje que indica que la persona está realizando tareas de limpieza.

Clase Profesorado

Esta clase tiene características adicionales específicas para los profesores:

Atributos específicos:

materias: Una lista de materias que enseña el profesor.

cargoDirectivo: Cargo directivo que puede ocupar, como director o jefe de estudios.

Método trabajar: Devuelve un mensaje detallado sobre las materias que enseña.

generarAlAzar: Crea un objeto Profesorado con una materia y cargo aleatorios.

__toString: Incluye información detallada sobre las materias y cargos directivos del profesor.

Clase AlumnoESO

Esta clase representa estudiantes de la Educación Secundaria Obligatoria (ESO). Los atributos adicionales incluyen:

curso: Curso en el que se encuentra el estudiante.

grupo: Grupo al que pertenece (por ejemplo, A, B, C, D).

Clase AlumnoBachillerato

Atributos adicionales: Similar a AlumnoESO, pero el atributo curso puede tomar solo los valores de 1 o 2, ya que en Bachillerato hay generalmente dos niveles.





Clase AlumnoFP

Atributos adicionales:

cicloFormativo: Nombre del ciclo de Formación Profesional que cursa el estudiante. curso y grupo: También definidos en AlumnoESO y AlumnoBachillerato.

Para cada una de estas clases de estudiantes, el método trabajar devuelve un mensaje indicando que están estudiando, adaptado a su nivel educativo y ciclo formativo en el caso de AlumnoFP.

PREGUNTA 2: (4,00 puntos)

Continuando con el supuesto de la pregunta 1, crea un script en PHP que:

- 1. Cree un array de 100 objetos al azar de los anteriores.
- 2. Indique cuántos objetos se crearon de cada clase.
- 3. Invoque para cada objeto el método trabajar().

Observaciones:

El script final ejecuta un bucle que crea 100 objetos de una clase seleccionada aleatoriamente. La clase específica se elige usando un número aleatorio entre 1 y 7, y, con base en este número, se crea una instancia de una de las clases derivadas (Administrativo, Profesorado, Conserje, etc.). Esto permite una generación de datos diversa y aleatoria para simular el personal y estudiantes en un contexto académico.

```
PS C:\Trabajos-DWC$\ud3-ejercicio!> php .\Persona.php
Kdebug: [Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9003 (through xdebug.client_host/xdebug.client_port).
Nombre: Maria Gómez López, DWI: 12345678T, Dirección: Calle Falsa 123, Sexo: M, Curso: 2, Grupo: B
Soy un estudiante de la ESO y estoy estudiando.
Nombre: Juan Gómez Rodríguez, DNI: 12345678R, Dirección: Calle Falsa 123, Sexo: M, Curso: 2, Grupo: A
Soy un estudiante de Bachillerato y estoy estudiando en el curso 2, grupo A.
 ombre: Luis Pérez Gómez, DNI: 12345678W, Dirección: Calle Falsa 123, Sexo: M, Años de servicio: 9
loy un administrativo y estoy gestionando tareas administrativas.
Nombre: Luis Rodríguez López, DNI: 12345678A, Dirección: Calle Falsa 123, Sexo: M, Ciclo Formativo: Mantenimiento Electrónico, Curso: 1, Grupo: C
Soy un estudiante de Formación Profesional en el ciclo Mantenimiento Electrónico y estoy en el curso 1, grupo C.
     bre: María Rodríguez Rodríguez, DNI: 12345678C, Dirección: Calle Falsa 123, Sexo: F, Años de servicio: 28, Materias: Historia, Cargo directivo: jefatura estudios personas adultas
una profesora y estoy enseñando la/s materia/s: Historia.
Nombre: María López Pérez, DNI: 12345678U, Dirección: Calle Falsa 123, Sexo: F, Años de servicio: 23
Soy una trabajadora de limpieza y estoy limpiando en el instituto.
Nombre: Juan Rodríguez López, DNI: 12345678N, Dirección: Calle Falsa 123, Sexo: M, Curso: 2, Grupo: B
Soy un estudiante de la ESO y estoy estudiando.
 Nombre: María Pérez Pérez, DNI: 12345678G, Dirección: Calle Falsa 123, Sexo: F, Años de servicio: 23
Soy una administrativa y estoy gestionando tareas administrativas.
 Nombre: Juan Gómez López, DNI: 12345678C, Dirección: Calle Falsa 123, Sexo: M, Años de servicio: 20, Materias: Ciencias, Cargo directivo: vicedirección
Goy un profesor y estoy enseñando la/s materia/s: Ciencias.
Nombre: Juan Gómez Pérez, DNI: 12345678J, Dirección: Calle Falsa 123, Sexo: M, Curso: 4, Grupo: C
Soy un estudiante de la ESO y estoy estudiando.
     bre: Juan Gómez Pérez, DNI: 12345678N, Dirección: Calle Falsa 123, Sexo: M, Curso: 1, Grupo: B
un estudiante de Bachillerato y estoy estudiando en el curso 1, grupo B.
Nombre: Luis Rodríguez Gómez, DNI: 12345678D, Dirección: Calle Falsa 123, Sexo: M, Años de servicio: 27
Soy un trabajador de limpieza y estoy limpiando en el instituto.
Nombre: Juan Gómez Pérez, DNI: 12345678P, Dirección: Calle Falsa 123, Sexo: M, Ciclo Formativo: Educación Infantil, Curso: 2, Grupo: C
Soy un estudiante de Formación Profesional en el ciclo Educación Infantil y estoy en el curso 2, grupo C.
 Nombre: Juan Rodríguez Gómez, DNI: 12345678P, Dirección: Calle Falsa 123, Sexo: M, Años de servicio: 1
Soy un conserje y estoy ayudando en el instituto.
 ombre: Ana Pérez López, DNI: 12345678H, Dirección: Calle Falsa 123, Sexo: F, Años de servicio: 29
oy una administrativa y estoy gestionando tareas administrativas.
      ore: María López López, DNI: 12345678D, Dirección: Calle Falsa 123, Sexo: M, Años de servicio: 36
un administrativo y estoy gestionando tareas administrativas.
```

















