

MATERIAL UTILIZABLE:

para la parte TEST, ningún material

para la parte práctica:

- todo el material de este módulo profesional y consulta a internet
 - la parte práctica se realizará sobre el entorno de desarrollo Visual Studio Code, disponible en una máquina virtual (se podrá elegir entre entorno en Windows o entorno en GNU/Linux Ubuntu).
- durante todo el tiempo del examen se grabará la pantalla con el software OBS previamente instalado.

PARTE 1: TEST**(2 puntos)**

Realizar el test en la plataforma FPADISTANCIA: 20 preguntas

Cada pregunta respondida correctamente en cada tema puntúa: 0,1 (= 2 puntos / 20 preguntas)

Cada pregunta respondida incorrectamente en cada tema puntúa: **-0,05** (= (-1/2) * 2 puntos / 20 preguntas)

La nota que muestra la plataforma al terminar el test NO es la correcta, pues esta plataforma no aplica los descuentos por fallos. La fórmula aplicar, para cada tema, es: nº de aciertos * 0,10 - nº fallos * (1/2) * 0,10

PARTE 2: PREGUNTAS PRÁCTICAS**(8 puntos)**

Criterios de puntuación:

- 1) CORRECCIÓN DEL CÓDIGO: **no importa en absoluto que haya mucho código o poco código**: lo que importa es que el código que desarrolles cumpla con las especificaciones indicadas en el enunciado, como mínimo
- 2) MEJORAS: cualquier prestación adicional que añadas, que no habiendo sido especificada explícitamente, pero no vaya contra las especificaciones indicadas, se puntuará positivamente.
- 3) LEGIBILIDAD del código (sangrado, espaciado de elementos, ...)
- 4) DOCUMENTACIÓN DEL CÓDIGO: comentarios correctos

Al terminar sube por favor en un único fichero .zip:

este fichero .odt con tus respuestas y observaciones, si las hay, a FPadistancia el código correspondiente al ejercicio PREGUNTA 1, todo dentro de una carpeta "unidad03"

PREGUNTA 1: (5,00 puntos)

Deseamos crear una aplicación que modelará o guardará información sobre las personas que conforman el personal de una empresa. En concreto, tenemos los siguientes tipos de personas, de los que queremos guardar la información:

para todos los trabajadores:

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

administrativos:

- años en la empresa
- cursos de formación completados (cada curso es una cadena de texto)
- sueldo

técnicos:

- años en la empresa
- certificaciones obtenidas (cada certificación es una cadena de texto)
- sueldo

repartidores:

- años en la empresa
- sueldo

gerentes:

- años en la empresa
- nivel
- sueldo
- plusporbeneficios

comerciales

- años en la empresa

- nivel
- sueldo
- plusporventa
- autonomos (son contratados externos a la empresa)
- disponibilidad
- preciohoraservicio
- rama
- formadores (son contratados externos a la empresa)
- disponibilidad
- preciohoraservicio
- rama
- personal de limpieza (son contratados externos a la empresa):
- años en la empresa
- sueldo

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 y que puede ser necesario añadir alguna clase intermedia.

implementa cada clase en un fichero separado.

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 **y cada vez distinta (dentro de lo posible)** (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 el rol de esa persona y lo que hace una persona de esa clase (por ejemplo: "Soy un autónomo y estoy arreglando las cañerías" ó "Soy una gerente y estoy en una reunión de gestión"; es decir, se tendrá en cuenta la propiedad "sexo" para imprimir un mensaje correcto).

Documenta todo el código.

PREGUNTA 2: (3,00 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, usando el método **numeroObjetosCreado()** y sin usar una estructura de datos externa.
3. Invoque para cada objeto el método **trabajar()**.