

Tema 3 – Introducción a la programación orientada a objetos

Parte II

Ejercicio 1

Crea la clase empleado con los siguientes atributos: nombre, edad, número empleado, salario. Utiliza tipos de datos apropiados para cada atributo.

Crea un constructor donde se inicialicen los atributos y otro constructor donde no se inicialicen.

Después crea los métodos get y set para poder modificar los atributos utilizando estos métodos.

Por último crea dos métodos que se llamen *plus* con los siguientes argumentos:

- El primer método plus tendrá como argumento la cantidad de dinero a incrementar sobre el salario que tiene. Si el empleado tiene más de 40 años entonces se realiza la operación de subida de salario.
- El segundo método tendrá como argumento la cantidad de dinero a incrementar y la retención que le hacen. Si el empleado tiene más de 40 años entonces se realiza la operación de subida de salario. La fórmula para el nuevo salario será:
 - $\text{Salario actual} = \text{salario actual} + \text{plus} - (\text{plus} * \text{retención} / 100)$
Donde el salario actual es el que tiene asignado actualmente el empleado, el plus es la cantidad de dinero que se le quiere incrementar al salario actual y retención es el % de retención que se le quiere hacer sobre ese extra.

Ejercicio 2

Realiza un programa que utilice el método random de la clase Math para relizar el juego de piedra, papel o tijera. Si sale un 1 será pidera, un 2 será papel y un 3 será tijera.

Ejercicio 3

En un almacén se guarda fruta para su posterior venta. Por cada cargamento se tiene la siguiente información: nombre de la fruta, procedencia, número de kilos, precio coste por kilo y precio venta por kilo. Codificar una clase para manejar esta información de forma que contenga las siguientes operaciones:

- Constructor
- Método que devuelva la información de cada cargamento de fruta.
- Método “rebajar” que rebaja el precio de venta en una cantidad pasada como parámetro, (el precio de venta nunca puede ser menor que el precio de coste).
- Método “vender”: se le pasa el número de kilos a vender y si hay suficiente cantidad, se decrementa el número de kilos y se devuelve el importe de la venta, sino da error.
- Método que nos diga si dos cargamentos de fruta tienen la misma procedencia.
- Llevar en todo momento el beneficio obtenido por el almacén.

Para probar dicha clase hacer un main que:

- Dé de alta 3 cargamentos y muestre su información.
- Diga si los dos primeros tienen la misma procedencia.
- Rebaje el precio del tercero.
- Realice ventas de los tres cargamentos.
- Muestre el beneficio obtenido por el almacén.

Ejercicio 4

En una agencia de viajes se guarda la siguiente información por cada viaje: Ciudad de origen y ciudad de destino, código, importe del viaje por persona, número de adultos, número de niños. Se pide hacer una clase Viaje que contenga al menos los siguientes métodos:

- Constructor. El código del viaje se forma con las dos primeras letras del origen (en mayúsculas), las dos primeras letras del destino (en mayúsculas) y el número de viaje.

Nota: hay que usar funciones de la clase String

- Método que reserve un número de plazas pasado como parámetro. Si no tiene plazas suficientes no reserva ninguna. Los niños tienen un 20% de descuento sobre el valor del billete. Devuelve el importe total de la reserva, 0 si no se puede realizar.
- Método que modifique el número de plazas de un viaje, sólo si es posible, es decir nunca puede haber más reservas que plazas. Retorna si se ha podido realizar o no la modificación.
- Método toString.
- Método que muestre el total recaudado por la agencia.

Codificar un main que realice las siguientes operaciones:

- Crear dos viajes.
- Mostrad la información de ambos viajes.
- Reservad varias plazas de ambos viajes.
- Modificad el número de plazas del primer viaje.
- Mostrad de nuevo la información de ambos viajes.
- Mostrad el total recaudado por la agencia.

Ejercicio 5

Por cada alumno del DAM guardamos su número de matrícula y sus notas en tres asignaturas. Hacer una clase alumno que guarde esta información. Realizar un programa que cree tres alumnos, les asigne notas y a continuación los muestre ordenados de mayor a menor nota media. Además, añadirle una variable que guarde el número de alumnos con nota media aprobada. Muestra después por pantalla el número de alumnos con nota media aprobada.

Ejercicio 6

Crea una clase Carro de la compra que contenga los datos del cliente: nombre, dirección y NIF. Al carro se podrán añadir productos y eliminarlos de él. De cada Producto se deben guardar el nombre, el precio base y el tipo de I.V.A. que se le puede aplicar (10%, 17% o 21% según sean productos básicos, ordinarios o de lujo).

Cuando el cliente decida realizar el pedido se mostrará por pantalla la lista de todos los productos, cada uno con su precio base y precio con I.V.A. así como el coste total.

Para las clases anteriores piensa qué criterios debe cumplir cada uno de los campos. En la clase principal (la que tenga el método main) comprueba si tu clase carro de la compra funciona correctamente. Prueba todo el programa haciendo que el método main añada varios productos al carro y luego muestre por pantalla la factura final.

Ejercicio 7

Vamos a crear una clase de billete de avión. Cada billete tendrá un nombre y un NIF de viajero que puede viajar en una de las tres clases: turista, business o VIP. Cuando llegue a facturar se le asignará un asiento determinado. Además, puede haber contratado el catering o no, el poder ir al baño o no. También se debe guardar el número de maletas que lleva.

Un cliente puede hacer una reserva inicial dando solo su nombre y nif o eligiendo una clase además. Prueba los constructores y métodos para verificar que funcionan correctamente.

Ejercicio 8

Modifica la clase anterior para establecer costes. Un billete en turista costará 200€, en business 400€ y en VIP 600€. Tanto business como VIP llevan el catering y el baño incluidos. Para los viajeros de clase turista que deseen contratar estos servicios, el catering costará 20€ y el baño 10€.

Además, la clase turista llevará una maleta incluida, business tres y VIP no tendrá límite en el número de maletas. Cada maleta adicional por encima del límite en la clase correspondiente costará 50€. El viajero podrá consultar en cualquier momento el coste total del billete.

Ejercicio 9

Implementa un método que recibe un número variable de parámetros de entrada de tipo String e imprima el número de parámetros recibidos y a continuación la posición del parámetro y su valor. Ejemplo:

Llamada al método:

```
pruebaParametros('Ana', 'Juan', 'Adrián');
```

Salida:

El número de parámetros es 3

Posición 0: Ana

Posición 1: Juan

Posición 2: Adrián

Ejercicio 10

Implementa un método que sume todos los parámetros (números enteros) que le lleguen (no se conoce de antemano el número de parámetros).

Ejercicio 11

Sobrecarga el método anterior de tal manera que reciba de forma fija 2 números reales (double o float) pudiendo venir a continuación más números enteros como parámetros o no. El método devolverá la suma de todos los parámetros que se le pasen en la llamada.

Ejercicio 12

Crea una clase llamada Calcula. Implementa los siguientes métodos estáticos, que devolverán en cada caso el menor o mayor número de los pasados como parámetros. En el caso de las cadenas devolverá la de mayor o menor longitud.

- *int mayor(int ... num)*
- *float mayor(float ... num)*
- *String mayor(String ... cadena)*
- *int menor(int ... num)*
- *float menor(float ... num)*

- *String* menor(*String ... cadena*)

Crea una clase aparte con el método *main* para probar todas las funcionalidades de la clase *Calcula*.