

## Programación en C#

### Actividad

1. (Clases) Crear una clase llamada Vehículo con los siguientes atributos: matrícula, marca (Toyota, Audi, Alfaromeo), tipo de vehículo (Todoterreno, Sedan, Deportivo), número de puertas (3, 5), precio base, llantas de alineación (si tiene o no), techo solar (si tiene o no) y número de airbags (hasta 6). Implementa, además de los constructores y las propiedades, los siguientes métodos:

- ✓ *precioTecho*: el precio se incrementará en un 2,3% si el vehículo dispone de techo solar.
- ✓ *precioPuerta*: si es un vehículo de tres puertas, el precio se decrementará un 0,5%.
- ✓ *precioMarca*: si el vehículo es de la marca Audi y, además, es un modelo deportivo, el precio se incrementará un 120%.
- ✓ *precioLlantas*: se incrementará el precio en 2.500€ si al vehículo se le añaden unas llantas de aleación.
- ✓ *precioAirbag*: el precio se incrementará un 1,5% por cada airbag que se añada.
- ✓ *importeTotal*: devolverá el precio final del vehículo con todos los extras.

Declara las constantes y enumerados que consideres necesarios para realizar el ejercicio. Además, haz en la clase los controles necesarios para que no se creen atributos con valores incorrectos (ejem: un coche con 2 puertas).

Crea una clase ejecutable en la que se creen y se muestre el importe final para al menos tres vehículos.

2. (Clases). Crea una clase Empleado con los siguientes atributos: nombre, antigüedad, si es extranjero, el número de horas que ha trabajado a la semana y su nivel (junior, senior o analista). Implementa, además de los constructores y las propiedades, los siguientes métodos:

- ✓ *salarioBase*: será el resultado de multiplicar el número de horas que el empleado está contratado a la semana por el precio al que se paga la hora (27 euros). En el salario base sólo se pagarán 40 horas a la semana como máximo.
- ✓ *horasExtra*: los trabajadores pueden trabajar un número de horas flexibles. Hasta 40 horas se pagará el salario base. A partir de las 40 horas se pagará un 30% más.

- ✓ *antigüedad*: cada vez que el empleado cumple un sexenio en la empresa recibe 90 euros más. Ejem: 4 años --> 0€. 6 años -> 90€. 12 años -> 180€. 14 años --> 180€.
- ✓ *extranjero*: a los empleados que son extranjeros, se le aumenta su salario un 12%.
- ✓ *nivel*: a los empleados de nivel senior se les aumentará su salario un 25% mientras que a los de nivel analista un 50%.
- ✓ *importeBruto*: será el resultado de la suma del salario base, horas extra, antigüedad, si es extranjero más su nivel.
- ✓ *importeNeto*: se calculará restando el impuesto al importe bruto.
  - ❖ 15% de impuesto si gana menos de 1400€.
  - ❖ 22% de impuesto si gana entre 1401€ y 1700€.
  - ❖ 27% de impuesto si gana más de 1700€.

Declara las constantes y enumerados que consideres necesarios para realizar el ejercicio. Crea una clase ejecutable en la que se muestre el importe neto de los siguientes empleados:

Apellidos, Nombre	Antigüedad	Extranjero	Horas	Nivel
Adams, Jackson H.	3	Yes	49	Analista
Bailey, Whilemina D.	11	No	42	Junior
Burnett, Kyla I.	7	No	46	Analista
Chapman, Eugenia Y.	8	Yes	37	Senior

3. (Clases). Haz una clase Persona que cumpla las siguientes condiciones.

- ✓ Sus atributos serán: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto.
- ✓ Se crearán varios constructores:
  - ❖ Un constructor por defecto.
  - ❖ Un constructor con el nombre, edad y sexo (el resto por defecto).
  - ❖ Un constructor con todos los atributos como parámetro.
- ✓ Los métodos que se deben implementar son:
  - ❖ *calcularIMC*: calculara si la persona está en su peso ideal (peso en  $\text{kg}/(\text{altura}^2 \text{ en m})$ ). Si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses constantes para devolver estos valores.

- ❖ *MayorDeEdad*: indica si es mayor de edad.
  - ❖ *comprobarSexo*: comprueba que el sexo introducido es correcto. Si no es correcto, será H. Este método no será visible al exterior.
  - ❖ *toString*: devuelve toda la información del objeto.
  - ❖ *generaDNI*: genera un número aleatorio de 8 cifras y a partir de este su número, se genera su letra correspondiente. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
- ✓ Ahora, crea una clase ejecutable que haga lo siguiente:
- ❖ Pide por teclado el nombre, la edad, sexo, peso y altura.
  - ❖ Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos para darle a los atributos un valor.
  - ❖ Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
  - ❖ Indicar para cada objeto si es mayor de edad.
  - ❖ Por último, mostrar la información de cada objeto.

**HAPPY  
PROGRAMMER  
DAY**