UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Industrial y de Sistemas

PRACTICA CALIFICADA I

Curso : **Programación orientada a objetos**
Código : **SI302V**
Alumno :

CICLO: 2021-I

Pregunta 1 (7 puntos)

Cree una clase que modele la información que una empresa mantiene sobre cada empleado: Nombre, Apellido, DNI, sueldo base, pago por hora extra, horas extra realizadas en el mes, tipo (porcentaje) de IR, estado civil y número de hijos. Use la encapsulación en los atributos de la clase.
Al crear los objetos se podrá proporcionar, si se quiere, el número de DNI.
Otras funcionalidades que deberán proporcionar los objetos de la clase serán los siguientes:

- Cálculo del monto correspondiente a las horas extra realizadas.
- Cálculo del sueldo bruto (sueldo base + monto por horas extras)
- Cálculo de las retenciones (Impuesto a la renta) a partir del tipo, teniendo en cuenta que el porcentaje de retención que hay que aplicar es el tipo menos 2 puntos si el empleado está casado y menos 1 punto por cada hijo que tenga; el porcentaje se aplica sobre todo el sueldo bruto.
- Método **visualizaEmpleado**(): imprime la información básica del empleado.
- Método **visualizaTodo**(): visualización de toda la información del empleado, es decir la información básica más el sueldo base, el monto por horas extra, el sueldo bruto, la retención de Impuesto a la renta y el sueldo neto.
- Sobrescriba el método "equals" de tal forma que se pueda decir que 2 empleados son los mismos si tienen el mismo número de DNI.

Ahora, cree una clase ejecutable que haga lo siguiente:

- Pida por teclado la cantidad de empleados y los datos de cada empleado.
- Permita validar si 2 empleados son iguales.
- Permita visualizar la información de cada empleado usando el método **visualizaTodo**():

Pregunta 2 (7 puntos)

Cree una clase "**CuentaBancaria**" con atributos para el número de cuenta (un entero largo), el DNI del cliente (otro entero largo), el saldo actual y el interés anual que se aplica a la cuenta (porcentaje). Defina en la clase lo siguiente:

- Un constructor por defecto y un constructor con DNI, saldo e interés
- Los atributos de la clase **CuentaBancaria** deben estar encapsulados
- Método **actualizarSaldo**(…): actualizará el saldo de la cuenta aplicándole el interés diario (interés anual dividido entre 365 aplicado al saldo actual).
- Método **ingresar**(double): permitirá ingresar una cantidad en la cuenta.

- Método **retirar**(double): permitirá sacar una cantidad de la cuenta (si hay saldo).
- Método **imprimirDetalle**(…) que nos permitirá mostrar todos los datos de la cuenta.
    - El número de cuenta se asignará de forma correlativa a partir de 100001, asignando el siguiente número al último asignado.
- Método **formateaDNI**(…) que nos permita mostrar el DNI formateado (ocho dígitos, un guion y la letra en mayúscula; por ejemplo: 41359635-T). La letra se calculará con un método auxiliar (privado) de la siguiente forma:
    - se obtiene el resto de la división entera del número de DNI entre 23 y se usa la siguiente tabla para obtener la letra que corresponde:

0 - T	1 - R	2 - W	3 - A	4 - G	5 - M	6 – Y
7 - F	8 - P	9 - D	10 - X	11 - B	12 - N	13 – J
14 - Z	15 - S	16 - Q	17 - V	18 - H	19 - L	20 – C
21 - K	22 – E					

- Se considera que dos cuentas bancarias son iguales si tienen el mismo número de cuenta y el mismo DNI del cliente.

Ahora, cree una clase ejecutable que haga lo siguiente:

- Genere "n" instancias de las cuentas bancarias. En cada cuenta genere al menos 2 ingresos y un retiro e imprima el detalle de las cuentas.
- Valide que no existan cuentas iguales.
- Muestre el saldo promedio de las cuentas, así como el DNI formateado del cliente que posee el mayor saldo en su cuenta (en el caso de la cuenta imprima el número de cuenta).

Pregunta 3 (5 puntos)

Jorge es un niño al que no le gustan las matemáticas así que nosotros le tenemos que enseñar cómo ejecutar las siguientes operaciones mediante programas sencillos:

3.1. Implemente el programa MCD que, en base a dos positivos, muestre el máximo común divisor entre ellos. El Algoritmo de Euclides es el siguiente:

* datos de entrada a y b positivos
* mientras b ≠ 0 repetir las tres instrucciones siguientes:
  r ← resto de a entre b (dar a r el valor del resto de a por b)
  a ← b (el nuevo valor de a es el antiguo valor de b)
  b ← r (el nuevo valor de b es el valor de r)
* el resultado es el último valor de a

Ejemplo:
Se busca el máximo común divisor de a = 945 y b = 651
945 = 1×651 + 294
651 = 2×294 + 63
294 = 4×63 + 42
63 = 1×42 + 21
42 = 2×21 + 0 entonces mcd(945; 651) = 21 (el último resto no nulo).

3.2. Calcule la función de Hackerman de forma recursiva

La fusión de Ackerman se define como:

Ackerman(m, n) = n + 1 si m = 0

Ackerman(m, n) = Ackerman(m − 1, 1) si m > 0 y n = 0

Ackerman(m, n) = Ackerman(m − 1, Ackerman(m, n − 1)) si m > 0 y n > 0

Con ello se tiene que Ackermann(1, 2) = 4 y Ackermann(3, 2) = 29

Suba el código fuente a Github (1 punto)

Suba la carpeta con el código fuente y el enunciado de esta práctica en un directorio privado en github llamado practica-1 y brinde acceso al profesor del curso.