
	Curso	DAM on-line			
	Módulo	PROGRAMACIÓN			
	Fecha	11 - Junio - 2020	Duración	120 minutos	
	Parcial	2 (JUNIO)	Temas	Del 6 al 9	
Criterios de calificación					Puntuación
Ejercicio Práctico: Parte obligatoria					8 puntos
Ejercicio Práctico: Parte a elegir					<u>2 puntos</u>
Total					10 puntos
Instrucciones generales					
El ejercicio práctico se compone de una parte común y una opción extra que se elegirá de las tres propuestas. La entrega del ejercicio práctico se realizara en un zip utilizando la plataforma avanza. El nombre del fichero será <i>apellido1_apellido2_nombre_PROG_2Parcial</i> y contendrá el proyecto realizado en netbeans.					

Nombre	PC
---------------	-----------

EJERCICIO PRÁCTICO: **Sastrería**.10 puntos

IMPORTANTE:

- **No** será necesario **validar** los datos. La especificación de los atributos se realiza para la elección del tipo de dato.
- **No** es necesario **comprobar que ya exista** un objeto con ese **nif** o con ese **código**.
- **No** es necesario aplicar un control riguroso de las **excepciones**.
- De la **opción extra** del ejercicio solo debes implementar una.

Nos han encargado hacer una pequeña aplicación para gestionar el sueldo de los empleados de una sastrería.

Clases en java:

- La clase **Persona**, tendrá los siguientes atributos: **NIF**, **nombre** y **apellidos**. Los métodos de la clase serán los métodos **set** y **get** para cada atributo y el método **toString**.
- La clase **Empleado**: esta hereda de **persona** y además de los atributos anteriores tendrá **codigo** y **sueltoBase**.
 - **codigo**, de empleado, con el siguiente formato **XX-X** siendo **X** un dígito. Ejemplo 10-1.
 - **sueltoBase**, sueldo base del empleado una cantidad entera. Ejemplo: 900
 Como método propio tendrá **calcularSuelto**, que se implementará en sus clases hijas.
- La clase **Sastre**: hereda de **Empleado** y además de los atributos anteriores tendrá:
 - **preciohora**, una cantidad entera. Ejemplo: 11.
 - **horastrabajadas**, va contabilizando las horas trabajadas, una cantidad entera. Ejemplo 6.
 El método **calcularSuelto** en este caso se calcula multiplicando las **horastrabajadas** por el **preciohora** y le sumamos el **sueltoBase**. Es decir $(\text{horastrabajadas} * \text{preciohora}) + \text{sueltoBase}$
- La clase **Vendedor**: hereda de **Empleado** y además de los atributos anteriores tendrá:
 - **turno**, que puede ser Mañana o Tarde.
 - **comision**, será un número entero que indica un porcentaje fijo (1-100), que se aplicará sobre las ventas realizadas. Ejemplo: 2 que indica un 2%.
 - **ventas**, un número con decimales que va contabilizando el total de ventas de un vendedor.
 El método **calcularSuelto** en este caso se calcula obteniendo la parte que le corresponde de **comisión** por ventas $(\text{comisión} * \text{ventas}) / 100$ le sumamos el **sueltoBase**. Es decir $((\text{comisión} * \text{ventas}) / 100) + \text{sueltoBase}$

Programa Principal: Crearemos una única **estructura de almacenamiento de datos** para guardar objetos de tipo **sastre** y **vendedor**. Está se gestionará con el siguiente **menú de operaciones**, en modo consola:

1. **Añadir empleado:** Esta opción nos debe pedir el tipo de trabajador y los datos para darle de alta. Se crea el objeto y se añade a una lista de empleados.
2. **Listar:** devolverá todos los datos de los empleados ordenados por código de empleado.
3. **Añadir venta:** Esta opción nos solicitará el código del empleado y nos permitirá añadir la nueva venta.
4. **Añadir trabajo:** Esta opción nos solicitará el código del empleado y nos permitirá añadir horas trabajadas.
5. **Calcular sueldo:** Esta opción nos solicitará un **nif** y mostraremos el sueldo que el corresponde a esa persona.

Opción extra a elegir (solo tienes que hacer una de las tres propuestas):

a.- **Guardar la lista de objetos en un fichero.** Para garantizar la persistencia de los datos. Esta opción debe guardar la lista y cuando se inicie la aplicación cargaría la lista en la aplicación. (El fichero será datos.dat)

b.- **Modo gráfico.** Implementar la **opción 3, añadir venta**, en modo gráfico para que nos permita introducir el código del vendedor y la venta, actualizándose los datos para ese objeto de la lista.

c.- **Guardar empleado-sueldo en una base de datos:** Esta opción se debe conectar con la base de datos sastreria, crea la tabla nomina con los campos **nif y sueldo** e insertamos esa información de todos los empleados.

Ejemplos de ejecución

(Opción 1 – Añadir empleado): Introduce el NIF: 12345678A Introduce el nombre: Pedro Introduce los apellidos: Marcos Sanchez Introduce el código del empleado: 01-1 Introduce el sueldo base del empleado: 900 Selecciona tipo de empleado: 1. Vendedor. 2. Sastre. 1 Introduce el TURNO del vendedor: Tarde Introduce la COMISIÓN del vendedor: 2	Opción 3 – Añadir Venta Introduce el código del Vendedor 01-1 Introduce el valor de la venta 244 Opción 4 - Añadir Trabajo Introduce el código del Sastre: 02-2 Introduce las horas de un trabajo: 6 Opción 5 - Calcular Sueldo Introduce el nif del empleado 87654321Z Sueldo Total:856
Opción 2 Listar Empleados Vendedor{01-1, 12345678A, Pedro Marcos Sanchez, 900, turno=Tarde, comision=2, ventas=244} Sastre {02-1, 87654321Z, Sara Jimenez Serrano, 850, preciohora=1, horastrabajos=6}	

Criterios de puntuación
<i>0,5 puntos – Clase Persona</i>
<i>0,5 puntos – Clase Empleado</i>
<i>0,5 puntos – Clase Sastre</i>
<i>0,5 puntos – Clase Vendedor</i>
<i>1 punto – Menú de operaciones</i>
<i>1 puntos – Alta</i>
<i>1 punto – Listar</i>
<i>1 punto – Añadir venta</i>
<i>1 punto – Añadir trabajo</i>
<i>1 puntos – Calcular sueldo</i>
<i>2 puntos – Opción extra a elegir</i>
NOTA