Detalles de la tarea de esta unidad.

Enunciado.

A lo largo de esta unidad has ido aprendiendo a crear tus propias clases así como sus distintos miembros (atributos y métodos). Has experimentando con la encapsulación y accesibilidad (modificadores de acceso a miembros), has creado miembros estáticos (de clase) y de instancia (de objeto), has escrito constructores para tus clases, has sobrecargado métodos y los has utilizado en pequeñas aplicaciones. También has tenido tu primer encuentro el concepto de herencia, que ya desarrollarás en unidades más avanzadas junto con otros conceptos avanzados de la Programación Orientada a Objetos.
Una vez finalizada la unidad se puede decir que tienes un dominio adecuado del lenguaje Java como para desarrollar tus propias clases y utilizarlas en una aplicación final que sea capaz de manipular un conjunto de datos simple. Dada esa premisa, esta tarea tendrá como objetivo escribir una pequeña aplicación en Java empleando algunos de los elementos que has aprendido a utilizar.

Se trata de desarrollar una aplicación Java denominado **PROG05_Tarea** que permita gestionar hasta 5 clientes con sus datos, esto se realizará mediante un menú con las siguientes opciones

1. **Nuevo Cliente.**
2. **Consulta de un cliente por DNI.**
3. **Consulta de un cliente por Apellidos y Nombre**
4. **Listado de todos los Clientes.**
5. **Salir.**

La funcionalidad será la siguiente:

- Al iniciar la aplicación se mostrará el menú propuesto.
- Dependiendo de la opción seleccionada por el usuario:

Nuevo Cliente: Se creará un nuevo Cliente, que contendrá la siguiente información (dni, nombre, apellidos, dirección, población, código postal). Todos los datos serán solicitados por teclado llamando a un método en la clase principal denominado **entrada_datos()**, este método devolverá un objeto cliente con los valores introducidos desde el teclado en dicho método, este objeto se asignará a uno de los **5 objetos clientes declarados como globales (cli1, cli2, cli3, cli4, cli5)** en la clase principal, para ello se pedirá antes de la entrada de datos del cliente el número al que le vas a asignar los valores (1,2,3,4,5), puede que si ese cliente ya tenga datos y por lo tanto lo que se estés haciendo es una modificación de los valores que ya tiene).

- Las comprobaciones que se tienen que realizar en este caso son las siguientes:

Que el DNI del cliente es correcto, método interno de la clase cliente (número de dígitos y letra asignada).

Si no se cumple algunas de las condiciones se deberá mostrar el correspondiente mensaje de error. En ese caso habrá se mostrará de nuevo el menú principal.

Consulta de cliente por DNI: Se pedirá en otro método llamado **consulta_dni(),** el dni del cliente a consultar, teniendo que ver si ese dni es el de uno de los 5 clientes existentes, esta comprobación se realizará desde un método en la clase Clientes llamado **existe_Dni(),** este método devolverá un valor booleano indicando si el dni buscado está entre los cinco clientes, si lo es se sacarán sus datos usando el método **toString()** de la clase Cliente en el método en la clase principal.

Consulta de cliente por Apellidos y Nombre: Se pedirá en otro método llamado **consulta_nombre()** los apellidos y el nombre del cliente a consultar, teniendo que ver si dos datos coinciden con alguno de los 5 clientes existentes (No se tendrán en cuanta las mayúsculas o minúsculas), esta comprobación se realizará desde un método en la clase Clientes llamado **existe_nombre(),** este método devolverá un valor booleano indicando el nombre y los apellidos buscados están entre los 5 clientes, si es el de alguno se sacará sus datos mediante el método **toString()** de la clase Cliente en el método de la clase principal.

Listado de todos los Clientes: Se llamará a un método en la clase principal listado_Clientes() en el que se mostrarán los datos de todos los clientes.

Salir: El programa finalizará.

El proyecto de Netbeans o Eclipse constará :

- **PROG05_Ejerc1**: que contendrá la clase **Cliente** y la clase **Principal**, una **clase** con **métodos estáticos** para realizar validaciones.

La clase `Cliente` dispondrá de los siguientes métodos:

- **Constructor o constructores.**

- **Métodos get y set para acceder a sus propiedades.**
- **Métodos de comprobación de búsqueda (existe_dni, existe_nombre)**
- **Método toString**

TEN EN CUENTA

- En la clase Cliente no debe solicitar datos por teclado ni escribir datos en pantalla. Esas operaciones se realizarán en la clase `Principal`.
- Debes incluir **una excepción** para la validación del DNI. Es decir, cuando no sea válido, se lanzará una excepción que se gestionará en la clase Principal, desde donde se mostrará el correspondiente mensaje de error.
- No será necesario realizar comprobaciones de tipo en los datos solicitados por teclado.
- No se podrán mostrar datos de un cliente si aún no se ha creado: en ese caso habrá que mostrar un mensaje por pantalla.

Piensa en los modificadores de acceso que debes utilizar tanto en atributos y métodos de la clase.

Además del programa deberás escribir también un informe con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea.

IMPORTANTE:

- En la cabecera de las clases añade documentación indicando autor y descripción de la clase.

No se podrán utilizar estructuras estáticas o dinámicas para almacenar los clientes (Arrays, Arraylist, TreeSet).

- En la cabecera de cada método añade documentación indicando la funcionalidad que implementa y el valor que devuelve.
- El código fuente Java de esta clase debería incluir **comentarios** en cada atributo (o en cada conjunto de atributos) y método (o en cada conjunto de métodos del mismo tipo) indicando su utilidad.

MEJORA

- Una mejora consistiría en, cuando un dato solicitado no es correcto, mostrar un mensaje y volver a solicitarlo hasta que se introduzca correctamente.

Se deben entregar el proyecto de Netbeans o Eclipse completo.

Criterios de puntuación. Total 10 puntos.

Para poder empezar a aplicar estos criterios es necesario que la aplicación compile y se ejecute correctamente en un emulador. En caso contrario la puntuación será directamente de 0,00.

Criterios de puntuación.

La clase Principal dispone de todos los **métodos** necesarios y funcionan.	3,00
La clase Cliente dispone de al menos un **constructor, los demás métodos** y funciona correctamente.	2,00
Opción 1 del menú.	1,50
Opción 2 del menú	1,00
Opción 3 del menú.	1,50
Se tienen en cuenta las excepciones	1,00
Total	**10,00**

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- JDK y JRE .
- Entorno de desarrollo NetBeans o Eclipse.

Consejos y recomendaciones.

Para realizar la aplicación te realizamos la siguiente serie de recomendaciones:

- Básate en los diferentes ejemplos que has tenido que probar durante el estudio de la unidad. Algunos de ellos te podrán servir de mucha ayuda, así que aprovéchalos.
- Puedes crear las clases que creas conveniente para llegar a una solución estructurada (por ejemplo, crear la clase DNI).
- El ejercicio resuelto de la clase DNI, en el cual se hacen comprobaciones de entrada, puede servirte de base para lanzar excepciones cuando no se cumplen ciertas condiciones. Además puedes utilizar ese código para realizar la validación el DNI del propietario del vehículo.
- En la Unidad 2, recuerda que hicimos una pequeña introducción al trabajo con cadenas de caracteres en Java.
- Si utilizas la clase Scanner, después de leer un entero lee una nueva línea para evitar errores de salto de línea. Es decir, para leer un entero siempre se deben ejecutar estas dos instrucciones:

```
valor=sca.nextInt();
sca.nextLine();
```

Indicaciones de entrega.

Una vez que tengas terminados el programa (fichero .zip de Netbeans y el documento explicativo, comprime ambos en un único archivo comprimido. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:
apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la quinta unidad del MP de PROG**, debería nombrar esta tarea como...
sanchez_manas_begona_PROG05_Tarea