6. UD6-PR.01: Almacenamiento de datos.

Detalles de la tarea de esta unidad.

En esta unidad, te enfrentas a la utilización de estructuras de datos de capacidad estática para almacenar información. Las estructuras explicadas en los contenidos son los arrays, tanto unidimensionales como multidimensionales. El término estático hace referencia a que su tamaño no puede variar en tiempo de ejecución.

También aprenderás la utilización avanzada de las cadenas de caracteres, incluyendo las expresiones regulares como una de las herramientas más potentes para el tratamiento de cadenas.

6.1 Gestión de vehículos: Taller<<Nombre>>

En la unidad anterior implementamos una aplicación capaz de gestionar un vehículo del taller Manolo. Evidentemente se trata de una aplicación muy limitada porque lo más lógico es gestionar un conjunto de vehículos, cada uno de ellos representado por una instancia de la clase Vehículo. Efectivamente, ese es el objetivo de la primera parte de la tarea de esta unidad: implementar la clase Taller***<<Nombre>>*** que será capaz de gestionar un máximo de 50 coches.

El proyecto se llamará PROG06_***<<Nombre>> donde*** **<<Nombre>> es tu nombre principal (en caso de tener varios, pon sólo 1).**

Mediante un menú que aparecerá en pantalla se podrán realizar determinadas operaciones:

- Nuevo Vehículo.
- Listar vehículos
- Buscar vehículo
- Modificar kms vehículo
- Salir.

El funcionamiento de cada opción será el siguiente:

- **Nuevo Vehículo**: Se solicitarán al usuario por teclado los datos de un nuevo vehículo y, si los datos son correctos, se almacenará en el taller. En caso de error en algún dato se volverá a solicitar hasta introducir un valor correcto.

  Para añadir un nuevo vehículo se deberá validar:

  - Mediante expresiones regulares que:
    - El DNI del propietario es correcto (tan solo el formato).
    - La matrícula del vehículo es correcta (tan solo el formato), es decir, tiene el formato NNNNLLL, donde NNNN es un número entre 0000 y 9999 y LLL son letras mayúsculas del abecedario.
  - Sin expresiones regulares (utilizando métodos de la clase String):
    - Que el nombre del propietario contenga al menos un nombre y dos apellidos (no tratar nombres compuestos) y su longitud no excede de 40 caracteres.
    - Habrá que comprobar que no existe en el taller un vehículo con la matrícula introducida. En caso afirmativo, se mostrará un mensaje por pantalla y mostrará el menú.

- **Listar Vehículos**: Mostrará por pantalla un listado de los vehículos del taller, mostrando toda la información disponible para cada vehículo.
- **Buscar Vehículo**: Se solicitará al usuario una matrícula por teclado (no será necesario validarla) y se buscará en el taller un vehículo cuya matrícula coincida con la introducida. Si existe se mostrarán su marca y matrícula por pantalla, y en caso contrario el mensaje "No existe vehículo con la matrícula introducida".

- **Modificar kms Vehículo**: Se solicitará al usuario por teclado una matrícula y un número de kilómetros. Si el vehículo con esa matrícula existe, se actualizará su número de kms al valor introducido. Si no existe, se mostrará un mensaje por pantalla.
- **Salir**: la aplicación finalizará.

Después de la ejecución de cada opción, se mostrará de nuevo el menú en pantalla.

6.2 CONSIDERACIONES A TENER EN CUENTA

- Puedes reutilizar la clase Vehículo que ya tienes implementada en la unidad anterior.
- Debes implementar la clase Principal que se encargue de:
  - Instanciar un objeto TallerX.
  - Pintar el menú y solicitar datos por teclado al usuario.
  - Realizar las validaciones de datos de entrada.
  - Mostrar datos por pantalla.
- Debes evitar el uso de la clase Vehiculo desde la clase Principal. Solo se debe utilizar la clase Taller<<Nombre>>.
- Debes implementar la clase Taller<<Nombre>> para aportar la funcionalidad que se especifica. Esta clase deberá contener la estructura de datos necesaria para almacenar los vehículos, con un tamaño máximo de 50. Por otro lado, ten en cuenta que para saber el número de vehículo existentes en la citada estructura, deberás utilizar un atributo de tipo entero. Este atributo te permitirá conocer la posición de inserción de un nuevo vehículo o hasta qué posición debes recorrer la estructura. Sus métodos serán:
  - **Constructor o constructores**.
  - **buscaVehículo**: Recibe como parámetro una matrícula, buscar el vehículo en el taller y devuelve una cadena con los datos del vehículo o null si el vehículo buscado no existe.
  - **insertarVehiculo**: Recibe todos los datos de un vehículo y trata de insertarlo en el taller. Devuelve 0 si se hizo con éxito, -1 si el taller está lleno y -2 si la matrícula ya existe.
  - **listaVehículos**: Lista por pantalla los datos de todos los vehículos del taller.

- **actualizaKms**: Recibe por parámetro una matrícula y un número de kilómetros, busca el vehículo con la cuya matrícula coincida y actualiza sus kilómetros. Devuelve true si se hizo con éxito y false en caso contrario.

- Utiliza los paquetes que consideres oportuno para organizar las clases.
- Ten en cuenta que los métodos de la clase Taller<<Nombre>> no deben mostrar datos por pantalla, a excepción del método que liste los vehículos. Estos métodos deben devolver un valor indicando si la operación se realizó correctamente o no.
- Piensa en los modificadores de acceso que debes utilizar tanto en atributos y métodos de la clase.

6.3 IMPORTANTE

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

6.4 TAREA OPTATIVA

- Implementar la opción de menú Eliminar vehículo, que solicitada una matrícula, elimine del taller el vehículo cuya matrícula coincida con la introducida por teclado.

6.5 Formato de entrega

Se deben entregar el proyecto de Netbeans completo. Para empaquetar un proyecto en Netbeans, utiliza la opción File - Export Project de Netbeans: generarás un fichero .zip con el contenido completo del proyecto.

Para la entrega de esta tarea debes generar un único fichero .zip/.rar que contenga todos los ficheros anteriormente indicados y que debes nombrar de la siguiente forma:

- Ciclo-módulo: DAM-PROG-
- número de unidad (UD6)
- identificación de la práctica (PR.01)
- apellidos, seguidos de tu nombre, separados por guión bajo.

"DAM-PROG-UD6-PR.01-apellido1_apellido2_nombre.xxx"

Ejemplo: DAM-PROG-UD6-PR.01-Fernandez_Lopez_Maria.pdf

Nota: Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas sería:

DAM-PROG-UD6-PR.01-Sanchez_Manas_Begona

6.6 Recursos necesarios para realizar la tarea

- Ordenador personal
- Sistema operativo Windows o Linux
- Conexión a Internet
- JDK y JRE de Java
- Entorno NetBeans

6.7 Consejos y recomendaciones

Para realizar la aplicación ten en cuenta las siguientes recomendaciones:

- Realizar pruebas a medida que escribes código.
- Comienza con la clase Principal y ve avanzando en la implementación de las clases y métodos que vayas necesitando.
- Crea expresiones regulares para detectar cuando se trata del DNI o NIE (esta la puedes

encontrar en los contenidos). No busques expresiones regulares complicadas, sino aquellas que de forma básica detecten de que se trata. En el siguiente enlace tienes contenidos que te pueden servir de ayuda con las expresiones regulares:

https://puntocomnoesunlenguaje.blogspot.com/2013/07/ejemplos-expresiones-regulares-java-split.html

6.8 Materiales

6.8.1 Recursos didácticos

- Apuntes en el aula virtual.
- Ordenador personal y conexión a internet
- Manual Java IES San Clemente: https://manuais.iessanclemente.net/index.php/Java
- Aprenda Java como si estuviera en primero (Universidad de Navarra): http://ocw.uc3m.es/cursos-archivados/programacion-java/manuales/java2-U-Navarra.pdf/view
- Libro "Thinking in Java 4th edition" Bruce Eckel (Disponible en recursos del módulo)