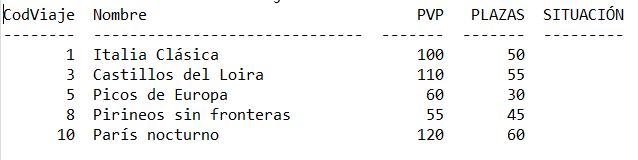
NOMBRE ALUMNO:

-----------------------------------------------------------------------------------------------------------------------------------------

En una empresa de viajes se dispone de un fichero con la información de los viajes que gestiona. El fichero se llama **Viajes.dat**, contiene la información de los viajes, su código, nombre, precio, y plazas ofertadas. Es un fichero aleatorio, con información guardada de forma aleatoria, donde **el campo identificativo de registro es el código de viaje**. En su creación no se ha utilizado el método writeUTF para las cadenas. La estructura es la siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cod Viaje**  **int** | **Nombre**  **String 30** | **Pvp**  **int** | **Plazas**  **int** | **Situación**  **String 1** |

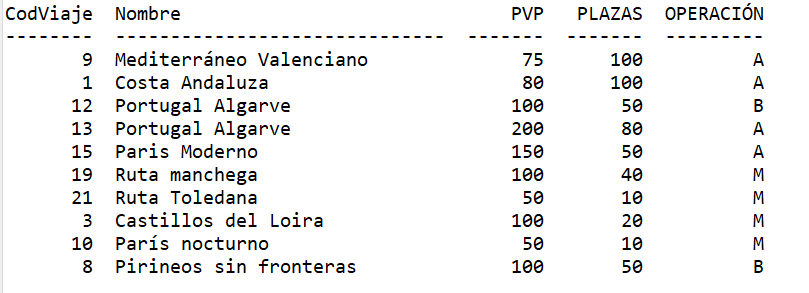
En situación inicialmente aparece un blanco. Nos indicará si el registro se ha actualizado o no. Inicialmente aparece en blanco, como que no se ha actualizado. El contenido del fichero es el siguiente:



La empresa va a añadir cambios a sus viajes, para ello ha creado un fichero aleatorio también, llamado Movimientos.dat. En este fichero los registros se han almacenado de manera secuencial. Sin ningún campo identificativo por registro. La información de este fichero es la siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cod Viaje**  **int** | **Nombre**  **String 30** | **Pvp**  **int** | **Plazas**  **int** | **Operación**  **String 1** |

El contenido es el siguiente:



Donde ***Operación*** indica la operación a realizar con el registro.

* Una ***A*** indica que el registro debe ser un **alta** en el fichero **Viajes**, es decir se debe de incorporar al fichero directo de viajes.
* Una ***B*** indica que el registro debe de ser una **baja**, es decir se ha de borrar del fichero **Viajes**. El borrado será lógico rellenando a 0 los numéricos, y a espacio en blanco los alfabéticos.
* Una **M** indica que es un viaje a modificar, es decir se ha de modificar el viaje en el fichero **Viajes**. La modificación consistirá en sumar al precio de los viajes el precio de movimientos y lo mismo con las plazas.

Al actualizar el fichero de **Viajes**, se indicará en el campo ***situación*** del fichero viajes, que el registro se ha actualizado (pondremos una ***M***), o borrado (pondremos una ***B***), o añadido (se pondrá una ***A***).

Ese campo lo utilizaremos para que sólo se pueda modificar una vez. Y podamos ejecutar varias veces la actualización sin que se produzcan más cambios de los necesarios en el pvp y las plazas.

A partir de estos dos ficheros se pide **crear un proyecto Maven, nombrarlo CON VUESTRO nombre+ape1+ape2,** que presente el siguiente menú y resuelva las siguientes operaciones:

OPERACIONES CON VIAJES

1. Actualizar Viajes.dat
2. Listar el fichero Viajes.dat.
3. Crear XML viajes.xml
4. Salir

Después de ejecutar cada operación se deberá mostrar el menú para poder ejecutar otra operación.

**Ejercicio 1:** **Actualizar el fichero Viajes.dat**.

Se debe de actualizar el fichero incorporando los cambios que vienen indicados en el fichero de Movimientos. Hay que controlar las situaciones de error, es decir que se vaya a dar de alta un viaje cuyo código de viaje ya exista. O que se vaya a modificar o borrar un viaje cuyo código de viaje no se localiza en el fichero Viajes.

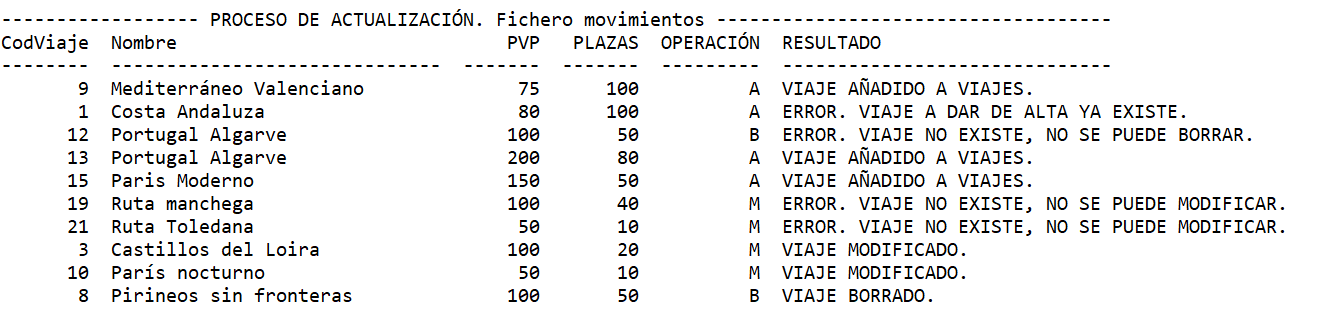
La modificación consistirá en actualizar los campos PVP y PLAZAS de Viajes, que se han de sumar con los de movimientos.

El alta consistirá en añadir el registro de movimientos en el fichero de Viajes.

La baja consistirá en un borrado lógico, poniendo a 0 o blancos los campos del fichero Viajes.

En **todos los casos** añadiremos al registro de **Viajes** en el **campo situación** la operación realizada (A, B, M)

Al realizar la actualización iremos obteniendo el siguiente listado:



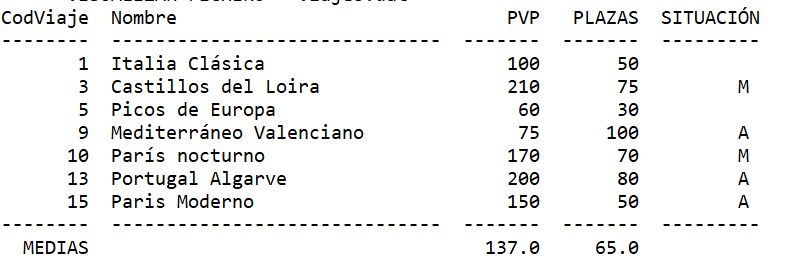
Si se vuelve a ejecutar la actualización no se deberán acumular los valores del pvp y de las plazas. Y los registros añadidos aparecerán como que ya existen.

El ejercicio se evalúa con 6 puntos:

* Visualizar correctamente los registros de movimientos en el momento de la actualización indicando los mensajes de las operaciones (columna resultado). (2 puntos)
* Realizar la actualización correctamente indicando la situación en el fichero Viajes.dat. 2 puntos.
* Controlar que las modificaciones de PVP y PLAZAS no se acumulen (1 punto)
* Mostrar salidas formateadas de manera tabular 1 punto.

**Ejercicio 2: Listar el fichero Viajes.dat**

Al pulsar esta opción se mostrarán los datos del fichero viajes con salida formateada:



El ejercicio se evalúa con 1,5 puntos. 0,5 mostrar los datos del fichero. 0, 5 cálculo de las medias correcto. 0,5 mostrar salida formateada correctamente.

**Ejercicio 3:** Crear fichero XML, viajes.xml.

Se desea obtener también un fichero XML de nombre **viajes.xml**, con la información de todos los viajes. El fichero tendrá la siguiente estructura por cada viaje:

<ListaViajes>

<Viaje>

<codviaje>xxxxxxxxxx</codviaje>

<nombreviaje>xxxxxxxxxxxxxx</nombreviaje>

<precio>xxxx</precio>

<numplazas>xxxxxxxxx</numplazas>

<operacion>xxxxxxxxx</operacion>

</Viaje>

<Viaje>

<codviaje>xxxxxxxxxx</codviaje>

<nombreviaje>xxxxxxxxxxxxxx</nombreviaje>

<precio>xxxx</precio>

<numplazas>xxxxxxxxx</numplazas>

<operacion>xxxxxxxxx</operacion>

</Viaje>

. . . . . . . .

</ListaViajes>

El ejercicio se evalúa con 2,5 puntos.

Creación de clases correctamente 1,25. Llenado de datos del fichero correcto 1,25.

**El ejercicio se evalúa con 0 puntos si se entrega con errores de compilación, o si salen errores de ejecución. Controlar las excepciones que surjan.**

**Una vez terminado el examen empaqueta el proyecto, pon tu nombre al empaquetado nombre+ape1+ape2, y súbelo a la plataforma al enlace indicado para ello.**

**El ejercicio penaliza 1 punto si el nombrado del proyecto y la entrega no se hace como se pide.**