CÓDIGO COMENTADO DE PROGRAMASERIES

Se realiza un programa en JAVA que gestiona un programa para incluir series a través de 3 *array*, estos recogen SERIE, PLATAFORMA Y GÉNERO los cuales tendrán 20 posiciones como máximo.

Se generan 3 *String* vacios con 20 posiciones disponibles que almacenaran las series, las plataformas y los géneros introducidos por el usuario.

Se utiliza *Scanner* para recibir las entradas del usuario

Se llama al método menú desde el programa inicial para que muestre por consola un rótulo de bienvenida al programa y las opciones disponibles del mismo. Añadir, modificar, eliminar, buscar y salir.

Posteriormente se llama al método inicializar desde el programa principal, este inicializará los datos recibiendo un *array* de *string*, contiene una variable longitud para calcular el tamaño del *array*, y evita que cada vuelta de bucle tenga que realizar una operación más. Se inicia con posiciones vacías "_".

A través de un bucle *do while* se procederá a ejecutar el programa según la opción recogida por el usuario a través de la plataforma. La elección del usuario debe ser alguna de las disponibles, si no el programa le indicará que "elija una opción del menú" por defecto.

Dichas opciones pueden ser:

- AÑADIR. Case 1. A través de un *if else*, si los datos introducidos, tanto de serie, plataforma o género supera la longitud del *array*, se le indicará al usuario que "el *array* está completo". Si no, se llama al método buscar de la clase buscador para encontrar una posición dentro del array. Se le pide al usuario que indique el nombre que tomará la posición vacía a través de plataforma, este quedará grabado y se imprimirá al final mostrando el contenido actualizado.

- MODIFICAR. Case 2. Con try catch, recogiendo este último una *exception* genérica que mostrará un mensaje de error, el usuario puede elegir qué entrada desea modificar y luego ingresar la nueva entrada. El método buscar de la clase buscador se utiliza para encontrar la posición de la entrada. Si la entrada realizada por el usuario no se encuentra, se imprimirá por pantalla un mensaje de error. Los datos actualizados se imprimirán al final.

- ELIMINAR. Case 3.Gestiona la opción de eliminar posiciones de los *array* con *try catch*. Se muestra por pantalla el contenido y se pide al usuario qué posición desea eliminar, si la entrada no se encuentra dentro de las posiciones mostradas se imprime un mensaje de error. Si es encontrada se llama al método privado *delete* para eliminar la posición completa. Se realiza de forma completa porque se entiende que el usuario querrá eliminar la serie completa. Se deja la posición libre con *longitud* --. Finalmente se imprimen los datos actualizados.

- BUSCAR. Case 4. En este caso se gestiona la opción de buscar una entrada dentro del array existente. Con *try catch*, se pide al usuario que introduzca la serie que busca, se llamará al método buscar de la clase buscador para encontrar dicha serie dentro del *array* actual. Si es encontrado se indicará en qué posición se encuentra la coincidencia. Como posición se inicia en -1 si se produce ese resultado se entiende como error (no encontrado), se indica por pantalla para informar al usuario que -1 indica que la entrada no se ha encontrado. Se muestra por pantalla los *array* existentes.

- SALIR. Case 5. Esta última opción gestiona la salida del programa. Se indica al usuario que ha salido del mismo.