



TAREA DE PROG05

Curso 2020-21

ENUNCIADO

- A lo largo de esta unidad se han explicado conceptos sobre clases, así como sus distintos miembros (atributos y métodos). Se ha experimentado con la encapsulación y accesibilidad (modificadores de acceso a miembros), se ha trabajado con miembros estáticos (de clase) y de instancia (de objeto), se han escrito constructores para las clases, se han sobrecargado métodos y se han utilizado en pequeñas aplicaciones. También se ha tenido un primer encuentro el concepto de herencia, que se desarrollarán en unidades más avanzadas junto con otros conceptos avanzados de la Programación Orientada a Objetos.
- La tarea no se dará por entregada si el proyecto tiene errores de sintaxis, no compila, no se ejecuta correctamente, si lo entregado no se corresponde con lo pedido o si no se ha realizado un mínimo de la tarea, es decir, si está vacía o casi.**
- En esta tarea hay que entregar:
 - Un proyecto en NetBeans llamado **Tarea_Prog05_Apellido1Apellido2Nombre** (Nombre es vuestro nombre y primer apellido), en el que se realizará las clases Java que se enumeren mas abajo.
 - Un documento **PDF** llamado: **Tarea_Prog05_Apellido1Apellido2Nombre.pdf**, en el que pondremos **captura de pantallas** en la que aparezca como **fondo** vuestra conexión **al Papas** y como primer plano las capturas del código fuente y los resultados de la ejecución del ejercicio (tal y como se vería con Netbeans), para así mostrar que funcionan (debe aparecer una prueba ejecutada de cada ejercicio). También realizaremos una explicación de lo que has realizado en la tarea.

Clase Libro

Esta clase debe incluir todas las herramientas necesarias para almacenar y trabajar con la siguiente información sobre los libros vendidos por una librería

- Atributos:**
 - Código del artículo: se almacenará mediante 3 atributos:
 - estanteria:** estantería donde están almacenado el libro (2 dígitos numéricos), almacenado como cadena (String).
 - balda:** balda donde está el libro (3 dígitos numéricos), almacenado como cadena (String)
 - numeroLibro:** número individual del libro (4 dígitos numéricos), almacenado como cadena (String).
 - Los dígitos de control no se almacenan porque se pueden calcular.
 - titulo:** título del libro (cadena de 40 caracteres)
 - unidadesTotales:** Unidades totales que hay de ese libro en el almacén (entero de 5 caracteres).
 - ACLARACIÓN: no hay que declarar el atributo CLIBC dentro del objeto, solo la estantería, balda y numeroLibro.
- Métodos Privados (private):**
 - separarCLIBC:** le pasaremos como entrada una cadena con el CLIBC (código de LIBRO completo), separará en los 3 atributos: almacen, balda y nº de libro y los almacenará en los atributos del objeto.
 - setTitulo:** pasándole como entrada una cadena con el título lo almacenará en el atributo **titulo**.
- Constructor:** le pasaremos el CLIBC y el título y llamando a los métodos **separarCLIBC** y **setTitulo**, asignará el valor a los atributos. Las unidades totales se inicializarán a 0.
- Constantes (final static):** tendremos 4 constantes: DIGITOS_ESTANTERIA: valor 2, DIGITOS_BALDA: valor 3, DIGITOS_NRO_LIBRO: valor 4, DIGITOS_DC: valor 2. Se utilizarán donde se necesite en lugar de usar el valor directamente.

- **Métodos estáticos y públicos (public static):**

- **calcularDigitosControl:** a este método se le pasarán 3 cadenas (String) como entrada estantería, balda y códigoLibro y nos devolverá los dígitos de control como valor entero. Para calcular los dígitos de control, convertiremos las cadenas en enteros (**Integer.parseInt**) y una vez convertidas sumaremos: estantería+balda + nº libro, se divide por 99 y se coge el resto. El resto son los dígitos de control.

Por ejemplo, si estantería tiene el valor: 01, balda tiene el valor : 002 y nº de libro tiene el valor 0001, sumamos $1+2+1=3/99=3$, el método devolverá: 3 como entero

- **correctoDigitosControlCLIBC:** le pasaremos como entrada al método un CLIBC y nos devolverá true si los dígitos son correctos y false si no lo son.

Mediante el método **CLIBC.substring(inicio,final)**, sacaremos del parámetro de entrada CLIBC los siguientes valores: estantería: 2 primeros caracteres, balda: 3 siguientes caracteres, nº libro: 4 siguientes caracteres y los dígitos de control: 2 siguientes caracteres.

Los pasamos a entero mediante **Integer.parseInt**.

Llamaremos al método **calcularDigitosControl**, pasándole los valores obtenidos: estantería, balda, nº libro, nos devolverá los dígitos de control como valor entero.

Se comparan los dígitos de control obtenidos del método **calcularDigitosControl** con los dígitos de control anteriormente obtenidos de CLIBC, si son iguales, devolverá true, si no lo son, false.

Por ejemplo, si CLIBC se le pasa el valor: 01002000105, sacaremos 01+002+0001, el dígito de control: 05, no es correcto ya que debería ser 4, devolverá false.

Si CLIBC tiene el valor: 01002000104, sacaremos 01+002+0001, el dígito de control introducido es 04, es correcto, devolverá true

- **correctoFormatoCLIBC:** le pasaremos una cadena CLIBC y nos devolverá true si es correcto (longitud =11 y que sea numérico) y false si no lo es.
- **correctoTitulo:** le pasaremos una cadena con el título del libro y nos devolverá true si el título es válido (longitud entre 1 y 40) y false si no lo es.

- **Métodos Públicos (public)**

- Obtener el valor de los atributos: **getEstanteria**, **getBalda**, **getNumeroLibro**, **getDigitosControl** (Llamará a método calcularDigitosControl) , **getTitulo**, **getUnidadesTotales**, **getCLIBC** (lo compondrá concatenando la estantería, la balda, el nº de libro y lo devuelto por el método calcularDigitosControl.
- **altaLibros:** se le pasa como entrada las unidades y se les sumarán a las unidades totales. Controlaremos que las unidades sean >0, en caso de no serlo levantaremos una excepción.
- **ventaLibros** se le pasa como entrada las unidades y se les restarán a las unidades totales. Controlaremos que las unidades sean > 0 y que sean < que las unidades totales en almacén, en cada caso levantaremos una excepción.

- Si crees conveniente introducir algún otro método, justifica su necesidad.

Clase GestionLibreria

En esta clase realizaremos lo siguiente dentro del método **main**:

- **Solicitar** al usuario que introduzca **por teclado** los siguientes datos: **Titulo** y Código del libro completo (**CLIBC**).
- **Titulo** del libro (con un máximo de 40 caracteres): Se introducirá por teclado el título del libro, para ello llamaremos al método **leerTeclado**, explicado mas abajo. A continuación llamará al método **Libro.correctoTitulo** que devolverá true si el título es válido (longitud entre 1 y 40) y false si no lo es. No continuará hasta que sea válido el título.
- **Código del Libro Completo (CLIBC)** es el código del libro completo. Se introducirá por teclado (llamando al método **leerTeclado**) un solo String, que será el código completo de 11 caracteres numéricos que estará compuesto por: la estantería (2 caracteres), balda (3 caracteres), número del libro (4 caracteres) y 2 dígitos control.

El programa deberá asegurarse que el **CLIBC** sea válido (no continuará hasta que lo sea), mediante la

comprobación de:

- El **formato** :11 dígitos: 2 dígitos de estantería, 3 dígitos para la balda, 4 para el número de libro y 2 dígitos de control. Llamaremos al método **Libro. correctoFormatoCLIBC**, al que le pasaremos la cadena introducida por teclado y nos dirá si es correcto o no (longitud debe ser 11 y que lo tecleado sea numérico).
- Los **dígitos de control** son válidos. Llamaremos al método **Libro. correctoDigitosControlCLIBC**, al que se le pasa CLIBC y comprueba que los dígitos de control sean correctos.

Si el CLIBC no es válido, no continuará, seguirá introduciendolo hasta que sea válido.

- Una vez que los datos introducidos sean correctos crearemos el objeto **libroActual**, pasándole al constructor CLIBC, el título del libro.
- Realizaréis un **menú** en el que se podrá elegir una de las siguientes opciones:
 1. **Ver el código del libro completo (CLIBC- Código Libro).**
 2. **Ver el título del libro.**
 3. **Ver la estantería.**
 4. **Ver la balda**
 5. **Ver el número del libro** (solamente el nº del libro).
 6. **Ver los dígitos de control del libro.** (se llamará al método `calcularDigitosControl` del objeto Libro)
 7. **Realizar altas de libros.** Habrá que solicitar por teclado las unidades que se desean dar de alta. Llamaremos al método **altaLibros**. Hay que recoger la excepción en caso de que se produzca algún error.
 8. **Realizar ventas de libros.** Habrá que solicitar por teclado las unidades que se desean vender. Llamaremos al método **ventaLibros**. Hay que recoger la excepción en caso de que se produzca algún error. No se pueden vender mas unidades de las que hay en almacén
 9. **Consultar unidades totales.**
 0. **Salir de la aplicación.**
- La visualización del Menu lo haremos introduciéndolo en un método llamado: **visualizarMenu** (será privado)
- Introduciremos una opción llamando al método **leerOpcion** (explicado después), realizaremos las sentencias para realizar cada una de las opciones y finalmente llamaremos al método **pulsarReturn** (explicado después), el programa se parará para poder ver el resultado de la opción elegida, pulsemos una Return y después volver a visualizar el menú.
- Dentro de la clase **GestionLibreria** crearemos los siguientes **métodos privados y estáticos**:
 - String **leerTeclado()**: en el que mediante `InputStreamReader` y `readLine`, nos devuelva un String con lo tecleado. Pondremos un try-catch y propagaremos la excepción, si salta alguna.
 - int **leerOpcion ()**: método por el que llamando al anterior nos devuelva una opción válida (entre 0 y 9). Pondremos un try-catch y propagaremos la excepción, si salta `NumberFormatException`, `IOException` o cualquier otra.
 - int **pulsarReturn()**: método en el introduciremos por teclado un carácter, lo usaremos para pulsar tecla Return y poder continuar con el programa.
- Utilizaremos el método **Integer.parseInt(String cadena)** para convertir una cadena en un entero.
- El código fuente Java debe 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.

CRITERIOS DE PUNTUACIÓN

Clase Libro : 5 puntos

Clase: GestionLibreria: 4 puntos

Aspectos generales : 1 puntos: presentación del informe , buen diseño proyecto, bien estructurado,etc.

RECURSOS NECESARIOS PARA REALIZAR LA TAREA.

- Ordenador personal.
- JDK y JRE de Java SE.
- Entorno de desarrollo Apache NetBeans.
- Antes de realizar la tarea descarga y mírate el ejercicio resuelto que se ha dejado en el apartado de Recursos Aportados por el Profesor: **Ejercicio Cuenta**. Lo puedes descargar en el siguiente enlace:
<https://aulasfp2021.castillalamancha.es/mod/resource/view.php?id=117316>

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.
- Ejercicio de ayuda: **Ejercicio Cuenta** (dirección descarga apartado anterior)

INDICACIONES DE ENTREGA.

- Lo que debes entregar:
 - Un proyecto en **NetBeans** llamado: **Tarea_Prog05_Apellido1Apellido2Nombre**
 - Además del proyecto crearemos un documento **PDF** llamado: **Tarea_Prog05_Apellido1Apellido2Nombre.pdf** en el que pondremos **captura de pantallas** en la que aparezca como **fondo** vuestra conexión **al Papas** y como primer plano con las capturas del código fuente y la ejecución del ejercicio (tal y como se vería con Netbeans), para así mostrar que funcionan (debe aparecer una prueba ejecutada del ejercicio).
- Comprimir el proyecto NetBeans y el documento pdf creados en un fichero llamado: **Tarea_Prog05_Apellido1Apellido2Nombre.zip**
- Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños
- En caso de tener que realizar un segundo envío, le daremos el siguiente nombre: **Tarea_Prog05_Apellido1Apellido2Nombre_ENVIO2.zip**