Algorítmica y Programación por Objetos 2
Ejercicio de Nivel 9
Tienda de libros

Descripción global

Se quiere crear un programa que permita administrar una tienda de libros. La tienda tiene un catálogo de libros, que son los libros que desea poner a la venta. La aplicación permite abastecer la tienda con ejemplares de los libros del catálogo y venderlos. Adicionalmente permite saber cuánto dinero se tiene en caja, empezando con una inversión inicial de $1.000.000.

Objetivos del ejercicio

El objetivo de este ejercicio es que el estudiante comprenda y adquiera práctica en:

- El desarrollo de aplicaciones siguiendo un proceso incremental.
- La implementación y uso de estructuras lineales sencilla y doblemente encadenadas.
- La creación de algoritmos de adición, eliminación y búsqueda en listas.
- El manejo de la clase Date para la manipulación de fechas.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

Preparación

Esta sección presenta una lista de chequeo de todas las tareas necesarias para la preparación del ejercicio. Por favor, revise que cada tarea haya sido completada **antes** de pasar a la siguiente sección de esta guía de trabajo.

Nota: En el siguiente enlace se encuentran las instrucciones de instalación de Java y Eclipse: https://cupitaller.uniandes.edu.co/instaladores/.

1. Para conocer el funcionamiento esperado de la aplicación, descargue y/o ejecute el archivo demo del ejercicio que se encuentra en el siguiente enlace: https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semestre-apo2/ejercicio-n9.

2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace: https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semestre-apo2/ejercicio-n9.

Universidad de los Andes | Vigilada MinEducación.
Reconocimiento como Universidad, Decreto 1297 del 30 de mayo de 1964 Personería Jurídica: Resolución 28 del 23 de febrero de 1949 MinJusticia.

3. Descomprima este archivo e importe el proyecto llamado **n9_tiendaDeLibros** a Eclipse. Limpie el ejercicio; para ello en Eclipse vaya a: **Project > Clean > Ok**. En el siguiente enlace hay un video con un ejemplo de cómo limpiar un proyecto en eclipse: https://youtu.be/mbcpY46wXS0.

4. Lea el enunciado del problema disponible en:
n9_ tiendaDeLibros/docs/specs/Descripcion.pdf.

5. Estudie el documento de requerimientos funcionales disponible en:
n9_ tiendaDeLibros/docs/specs/RequerimientosFuncionales.pdf.

6. Estudie el documento de consideraciones adicionales de diseño disponible en:
n9_ tiendaDeLibros/docs/specs/ConsideracionesAdicionalesDeDisenho.pdf.

7. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en: **n9_tiendaDeLibros/docs/specs/ModeloConceptual.png**. Identifique las clases, relaciones entre clases, constantes, atributos y métodos.

8. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Puede ver el tutorial en: http://cupitaller.virtual.uniandes.edu.co/videos-guia/.

Desarrollo

Parte 1: Creación del mundo

1. Descargue de la página de cupi2 el ejercicio de nivel 3 Tienda De Libros (https://cupi2.virtual.uniandes.edu.co/nivel-3/casos-de-estudio/tienda-de-libros).

2. Copie las clases Transaccion, Libro y TiendaDeLibros en el esqueleto de este nivel.

Parte 2: Clase Transaccion

En el archivo Transaccion.txt encontrará los contratos y signaturas de algunos de los métodos que debe completar de la clase Transaccion. Recuerde reemplazar los contratos y las signaturas de sus métodos con los proporcionados en el archivo.

1. Revise en el modelo del mundo la clase Transaccion, para identificar los nuevos atributos y métodos.

2. Declare e documente el atributo **siguiente**.

3. Modifique el atributo fecha para que sea de tipo **Date**.

4. Modifique el método constructor Transaccion para que el parámetro pFecha ahora sea de tipo Date y no String. Recuerde modificar el contrato para que se adapte a este cambio.

5. Modifique el método darFecha para que sea de tipo **Date**. Recuerde modificar el contrato para que se adapte a este cambio.

6. Copie la documentación y la signatura del método **darSiguiente** que se encuentra en el archivo Transaccion.txt. Implemente este método para que cumpla con la documentación.

7. Copie la documentación y la signatura del método **cambiarSiguiente** que se encuentra en el archivo Transaccion.txt. Implemente este método para que cumpla con la documentación.

Parte 3: Clase Libro

En el archivo Libro.txt encontrará los contratos y signaturas de algunos de los métodos que debe completar de la clase Libro. Recuerde reemplazar los contratos y las signaturas de sus métodos con los proporcionados en el archivo.

1. Revise en el modelo del mundo la clase Libro, para identificar los nuevos atributos y métodos.

2. Declare y documente el atributo **categoria**.

3. Elimine el atributo **transacciones** y declare y documente el atributo **primeraTransaccion**.

4. Declare y documente los atributos **anterior** y **siguiente**.

5. Modifique el método constructor para que cumpla con la declaración y la documentación que se encuentra en el archivo Libro.txt.

6. Copie la documentación y la signatura del método **darCategoria** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

7. Copie la documentación y la signatura del método **darAnterior** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

8. Copie la documentación y la signatura del método **darSiguiente** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

9. Copie la documentación y la signatura del método **darPrimeraTransaccion** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

10. Copie la documentación y la signatura del método **cambiarAnterior** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

11. Copie la documentación y la signatura del método **cambiarSiguiente** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

12. Copie la documentación y la signatura del método **agregarTransaccion** que se encuentra en el archivo Libro.txt. Implemente este método para que cumpla con la documentación.

13. En el método vender:

    a. Modifique el parámetro pFecha para que ahora sea de tipo **Date** y no String. Modifique el contrato para que se adapte a este cambio.
    b. Modifique la implementación del método para que se adapte a la nueva estructura de listas enlazadas, manteniendo las condiciones del contrato.

14. En el método abastecer:
    a. Modifique el parámetro pFecha para que ahora sea de tipo **Date** y no String. Modifique el contrato para que se adapte a este cambio.
    b. Modifique la implementación del método para que se adapte a la nueva estructura de listas enlazadas, manteniendo las condiciones del contrato.

15. Modifique el método darTransacciones para que itere en la lista enlazada de transacciones y retorne un ArrayList con todas las transacciones.

16. Copie el método compararPorTitulo que se encuentra en el archivo Libro.txt.

17. Modifique el método toString para que retorne una cadena con la siguiente estructura: <titulo> (<isbn>) - <categoria>.

Parte 4: Clase TiendaDeLibros

En el archivo TiendaDeLibros.txt encontrará los contratos y signaturas de algunos de los métodos que debe completar de la clase TiendaDeLibros. Recuerde reemplazar los contratos y las signaturas de sus métodos con los proporcionados en el archivo.

1. Revise en el modelo del mundo la clase TiendaDeLibros, para identificar los nuevos atributos y métodos.

2. Declare y documente la lista de constantes CATEGORIAS, para que incluya las categorías que se enlistan en el documento de descripción.

3. Elimine el atributo **catalogo** y declare y documente el atributo **primerLibro**.

4. Modifique el método constructor para que cumpla con la declaración y la documentación que se encuentra en el archivo TiendaDeLibros.txt.

5. Modifique el método **darCatalogo** para que itere en la lista enlazada de libros y retorne un ArrayList con todos los libros.

6. Copie la documentación y la signatura del método **darPrimerLibro** que se encuentra en el archivo TiendaDeLibros.txt. Implemente este método para que cumpla con la documentación.

7. Modifique el método **buscarLibroPorTitulo** para que se adapte a la nueva estructura de la lista enlazada de libros, manteniendo las condiciones del contrato.

8. Modifique el método **buscarLibroPorISBN** para que se adapte a la nueva estructura de la lista enlazada de libros, manteniendo las condiciones del contrato.

9. En el método **registrarLibro:**

    a. Remplace la documentación y signatura por la que se encuentra en el archivo TiendaDeLibros.txt.
    b. Implemente este método para que cumpla con la documentación y la nueva estructura de lista enlazada de libros.

10. Modifique el método **eliminarLibro** para que se adapte a la nueva estructura de lista enlazada de libros, manteniendo las condiciones del contrato.

11. Modifique el método **abastecer** para que se ajuste al nuevo método abastecer de la clase Libro.

12. Modifique el método **vender** para que se ajuste al nuevo método abastecer de la clase Libro.

13. Modifique el método **darLibroMasCostoso** para que se adapte a la nueva estructura de lista enlazada de libros, manteniendo las condiciones del contrato.

14. Modifique el método **darLibroMasEconomico** para que se adapte a la nueva estructura de lista enlazada de libros, manteniendo las condiciones del contrato.

15. Modifique el método **darLibroMasVendido** para que se adapte a la nueva estructura de lista enlazada de libros, manteniendo las condiciones del contrato.

Universidad de los Andes | Vigilada MinEducación.
Reconocimiento como Universidad, Decreto 1297 del 30 de mayo de 1964 Personería Jurídica: Resolución 28 del 23 de febrero de 1949 MinJusticia.

Parte 5: Modificación de la interfaz

En DialogoRegistrar completar los TODOs que comienzan por: **//TODO Parte 5 punto X**

Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

1. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz. Los resultados obtenidos deben ser iguales a aquellos mostrados en el video demo.

2. Ejecutar las pruebas automáticas disponibles en el ejercicio. En el siguiente video https://youtu.be/rVd4AD8XMJk se explica cómo efectuar esas pruebas. Estas pruebas deben presentar resultados en verde (0 errores y 0 fallas) cuando el ejercicio ha sido completado correctamente.

Tenga en cuenta que esas pruebas no son exhaustivas y que su correcto funcionamiento no garantiza que no haya ningún error en su programa.

Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Indente el código fuente de todas las clases del mundo. En el siguiente enlace https://youtu.be/cwQ9QiauaSc encuentra un video que explica cómo indentar el código fuente de su ejercicio.

2. Limpie el proyecto para que la entrega no contenga archivos ejecutables ni temporales (https://youtu.be/mbcpY46wXS0).

3. Construya el archivo entregable con su ejercicio desarrollado y validado completamente. En el siguiente video https://youtu.be/xuSDFfEZW78 se explica detalladamente el proceso para producir el comprimido del ejercicio y enviarlo a SicuaPlus. Renombre el archivo a entregar con su login de la siguiente forma:

   n<*nivel del ejercicio*>_<*login estudiante*>**.zip**
   Por ejemplo: **n9_tsuarez.zip**

   La no indentación del código fuente o el nombramiento incorrecto del ejercicio en su entrega es una acción penalizada en la plantilla de calificación del mismo.

4. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.