Proyecto - Aplicación de consola “libreta de direcciones”

Descripción del proyecto y requisitos

Objetivo y resultados:
Este proyecto le servirá para desarrollar sus habilidades de POO a través del desarrollo de una aplicación de consola “libreta de direcciones” que requiere ensamblar una solución a partir de clases modulares, aplicando las buenas prácticas para la construcción de software. Esto requiere:

- Aplicar "Divide and Conquer", una fortaleza del diseño POO que le permite usar la modularidad y la encapsulación para dividir un problema
- Desarrollar código de pruebas unitarias
- Usar Colecciones en Java para implementar Estructuras de Datos
- Crear una única instancia estática de AddressBook a la que accederán todas las demás clases y el código de cliente; en POO, esto a veces se llama patrón "singleton".
- **Aplicar buenas prácticas para lograr un buen código:**
  - **A salvo de errores, o en otras palabras correcto;**
  - **Fácil de entender para otros programadores; y**
  - **Listo para ser actualizado en el futuro.**

Clases:
Con el fin de orientar el desarrollo, a continuación, se describe una lista de clases que debe implementar y sus significados:
Los paquetes que usarás y sus clases son:

1. **address** = contiene la clase AddressBookApplciation y la clase Menu
2. **address.data** = contiene AddressEntry, AddressBook

- **AddressBook** - esta clase representa y contiene posiblemente cada "list" creciente y/o reducida de AddressEntries. Permite varias operaciones como buscar/encontrar, agregar y eliminar AddressEntries.
  - Debe pensar en la eficiencia de las operaciones necesarias --vea el ejemplo de ejecución –mostrar contactos en orden alfabético, agregar, eliminar, buscar, leer contactos de un archivo de texto

- **AddressEntry:** esta clase representa una sola entrada de información dirección/contacto en la libreta de direcciones
  - Debe contener constructores, al menos un constructor... Uno que acepte un ***nombre***, ***apellido***, ***calle***, ***ciudad***, ***estado***, ***código postal***, ***correo electrónico***, ***teléfono***.

- Contiene variables de clase independientes que representan toda la información de un contacto AddressEntry.
- Debe incluir un método toString() que devuelva un formato agradable de una cadena que contenga toda la información de contacto para imprimir en la consola
- Debe contener los métodos setX() y getX() donde X son las variables de clase. (por ejemplo, setName(***) y getName() donde *** son parámetros apropiados)

- **AddressBookApplication** - contiene la clase principal de la aplicación.
  - Utiliza las clases Menu y AddressBook como se describe en este documento
  - Aquí se llama al método displayMenu() de Menu, capturar las entradas del usuario y llamar adecuadamente a los métodos correctos

- **Menu**: esta clase se utiliza para mostrar las opciones de menú al usuario.
  - El método displayMenu() muestra las opciones a-f que se describen a continuación (ver también el ejemplo de ejecución al final de este documento):

    a) Carga de entradas desde un archivo.

    b) Agregar - solicitar al usuario información que se utilizará para crear un nuevo AddressEntry

    c) Eliminar - eliminación de un AddressEntry de la Agenda. Primero se realiza una búsqueda (ver ejemplo de ejecución) y luego el usuario selecciona de los resultados de búsqueda el contacto que desee eliminar.

    d) Buscar: solicita a los usuarios el inicio del apellido del usuario (pueden ingresar un apellido completo o solo el comienzo.... se mostrarán todas las entradas que coincidan con la primera parte de su apellido). Tenga en cuenta que en el caso de que se encuentre más de una entrada, se devolverán todas las coincidencias

    e) Mostrar - listado de las direcciones en orden alfabético por el apellido de la persona.

    f) Salir -tenga en cuenta que NO estamos guardando ningún objeto AddressEntry recién creado. Opcional. Implementar la persistencia de los datos

El siguiente **diagrama de clases UML** muestra las clases que creará y ALGUNOS (por ejemplo, falta el método displayMenu de la clase Menu) de los métodos que necesitará:

deberá actualizar este diseño para incluir todos los métodos que diseñe y cree para cumplir con los requisitos de este proyecto.

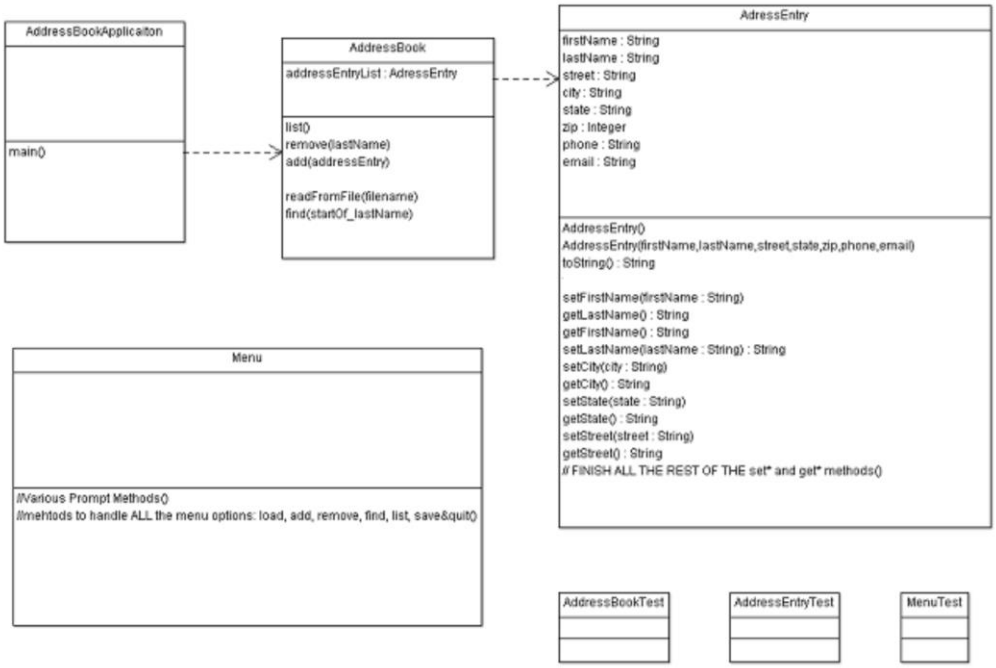

Figura 1. Diagrama UML

Pruebas unitarias:

Trabaje en sus pruebas unitarias antes de escribir demasiado profundo el código de la aplicación de la libreta de direcciones. Debe crear una clase de prueba para cada clase que cree. Por ejemplo, para la clase AddressBook cree una clase de prueba JUnit denominada AddressBookTest. Escribir las pruebas al principio (o casi) te ayuda a empezar a pensar en los métodos y los casos de uso antes de escribir el código. Tener las pruebas en su lugar hace que sea fácil ver si su código está funcionando a medida que crea el código para los distintos casos. Considere el código de prueba unitaria de la siguiente manera:

- Para una clase de prueba en particular, debe probar los métodos de su clase correspondiente al menos 2 veces para cada método. Por ejemplo, la prueba unitaria AddressBookTest debe revisar 2 o más instancias diferentes de un AddressBook usando su constructor, debe llamar y verificar la salida de cada uno

de sus métodos --addAddress, getAddress, equals, find, lessThan, mayorThan al menos 2 veces para cada método. Tenga en cuenta que en este caso AddressBookTest tendrá métodos de testAddressBook, testAddAddress, testGetAddress, testEquals, testFind, testLessThank, testgreaterThan, etc.

- Cuando se ejecutan contra una clase correcta, los métodos de cada clase de prueba unitaria no deben informar de ningún problema.
- Si alguien introdujera algún error en una clase, entonces la clase de prueba correspondiente debería mostrar el error. Su prueba unitaria debe mostrar que hay un problema, pero no necesita decir exactamente cuál es el problema.... Eso es lo que se hace con la depuración utilizando la información/indicación de fallos de prueba unitaria.

Especificaciones y código

- Antes de codificar, cree un documento UML que muestre su diseño.
- DOCUMENTA el código
- Piense en la eficiencia al elegir sus estructuras de datos y algoritmos. Puede asumir que los usuarios igualmente querrán buscar, almacenar en archivo, leer del archivo, enumerar alfabéticamente todas las entradas. Debe incluir en la descripción de su código por qué elige las estructuras de datos que hizo y cuál es la eficiencia para ordenar (como lo hizo), buscar / encontrar, enumerar alfabéticamente
- DEBE desarrollar casos de pruebas unitarias para todas sus clases y sus métodos importantes. Esto es parte de los entregables.
- Validación de entradas del usuario.
    - Cuando el usuario ingresa una AddressEntry, NO debe asumir que el usuario ingresa los datos correctos, probar para asegurarse de que un número sea un número, etc. De lo contrario, debe IMPRIMIR un mensaje diciéndole al usuario que es una entrada INVÁLIDA y volver a solicitar la entrada en cuestión. Solo tiene que verificar el tipo --- no necesita ver, por ejemplo, si el número de teléfono tiene una lada válida o si la dirección de correo electrónico es válida.
- Cree una clase denominada Address que contenga variables de instancia de cadena para calle, ciudad, estado y código postal.
    - Esta clase debe contener un constructor que acepte valores para inicializar estas variables y métodos que devuelvan cada una
    - Método toString() que devuelve una cadena para el objeto Address.
    - setX y getX métodos para establecer/obtener variables de clase. (es decir, setCalle, setCiudad, etc.)
- A medida que codifica, cree comentarios y produzca JavaDocs que documenten completamente TODAS sus clases y cargue sus archivos JavaDoc en su cuenta para que sean accesibles a través de GitHub.
- Pruebe su código y almacene los resultados en un documento.

Git

Se requiere crear un repositorio Git localmente y un repositorio remoto GitHub llamado TuApellido_LibretaDirecciones. Debe utilizar el repositorio para desarrollar su código.

DEBES comentar bien cualquier compromiso (commit) que hagas, se revisará este aspecto del commit (Se sugiere incluir el comentario en inglés).

Entregables:

Documento que integre las siguientes secciones:

- Publicar su código (y artefactos requeridos) en GitHub
- Elaborar documento (o wiki en GitHub) que integre los siguientes apartados

  - Sección A: Descripción: proporcione una descripción detallada de cómo funciona su sistema, que incluye:

    - A.1) Estado del sistema: una descripción de qué partes funcionan o no ---dar detalles. Si su sistema NO funciona y no muestra capturas de pantalla de cómo está fallando y muestra que a través de capturas de pantalla intentó insertar puntos de interrupción y terminó la depuración, se considerará.
    - A.2) Lista de clases: con una breve descripción del propósito, discuta cualquier estructura de datos / algoritmos utilizados su eficiencia en relación con las operaciones principales de nuestro menú.

  - Sección B: URL de JavaDoc: Publicación de código JavaDoc en su cuenta de GitHub. Cada clase, sus variables y métodos DEBEN ser documentados/descritos con la buena práctica aplicada de acuerdo a lo visto en clases.

  - Sección C: UML/Diseño: contiene lo siguiente

    - C.1) Diagrama de jerarquía de clases UML: muestra la jerarquía y la cardinalidad y las asociaciones de objetos para este proyecto.

  - Sección D: Capturas de pantalla del sistema: capturas de pantalla que muestran su aplicación funcionando para los siguientes casos:

    - D.1) Evidencia 1: Leer entradas del archivo de datos seguidas de Mostrar listado. El archivo de datos debe contener un mínimo de 2 direcciones. Incluir captura de pantalla de salida y también mostrar el contenido correspondiente del archivo de datos

    - D.2) Evidencia 2: inmediatamente después de D.1, agregue un nuevo objeto AddressEntry seguido de un mostrar el resultado de un nuevo listado en consola

    - D.3) Evidencia 3: inmediatamente después de D.2 hacer una eliminación de una entrada seguida de un mostrar listado

    - D.4) Evidencia 4: inmediatamente siguiendo D.3 hago una búsqueda usando una entrada que debería recuperar al menos una entrada. Ahora haga una búsqueda usando una entrada que no debería recuperar ninguna entrada. Coloque capturas de pantalla de cada hallazgo aquí.

- (Opcional) Sección E: Historial de Commits Capturas de pantalla que muestran las confirmaciones del repositorio de GitHub

3. Suba el documento, la URL de su repositorio y el código (Código.zip comprime todo el proyecto) a EMINUS (Actividad de Proyecto Final)

ANEXO I Ejemplo de la salida del programa

Sesión de muestra: el usuario elige elementos de menú sin ningún orden en particular. Notas: las entrada del usuario se muestra en rojo, y el estado inicial tenía 2 entradas en la Libreta de direcciones ---solo como ejemplo.

```
====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
e

1: José Camacho
Independencia 142
Xalapa, Ver. 96400
josec@gmail.com
928-4236652

2: Maria Flores
Juan Escutia 1022
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
b

Nombre:
Juan
Apellido:
Arias
Calle:
Revolución 22
Ciudad:
Minatitlán
Estado:
Veracruz
CP:
94233
Email:
juan@gmail.com
Teléfono:
922-8024552
```

```
====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
e

1: Juan Arias
Revolución 22
Minatitlán, Ver. cp. 94233
juan@gmail.com
922-8024552

2: José Camacho
Independencia 142
Xalapa, Ver. cp. 96400
josec@gmail.com
928-4236652

3: Maria Flores
Juan Escutia 1022
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
b

Nombre:
Norma
Apellido:
Perez
Calle:
Roble 18
Ciudad:
Minatitlán
Estado:
Veracruz
CP:
95000
Email:
norma@gmail.com
Teléfono:
922-66665555
```

```
==================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
==================================
e

1: Juan Arias
Revolución 22
Minatitlán, Ver. cp. 94233
juan@gmail.com
922-8024552

2: José Camacho
Independencia 142
Xalapa, Ver. cp. 96400
josec@gmail.com
928-4236652

3: Maria Flores
Roble 18
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

4: Norma Perez
Juan Escutia 1022
Minatitlán, Ver. 95000
norma@gmail.com
922-66665555

==================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
==================================
a

Ingresa el nombre del archivo:
datos.txt

==================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
==================================
```

```
e


1: Juan Arias
Revolución 22
Minatitlán, Ver. cp. 94233
juan@gmail.com
922-8024552

2: José Camacho
Independencia 142
Xalapa, Ver. cp. 96400
josec@gmail.com
928-4236652

3: Maria Flores
Roble 18
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

4: Norma Perez
Juan Escutia 1022
Minatitlán, Ver. 95000
norma@gmail.com
922-66665555

5: Manuel Soto
Allende 100
Coatzacoalcos, Ver. 96400
manuel@gmail.com
921-4444333

6: Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
c

Ingresa apellido del contacto a eliminar:
Soto

El siguiente contacto fue encontrado:
Manuel Soto
Allende 100
Coatzacoalcos, Ver. 96400
manuel@gmail.com
```

```
921-4444333

Ingresa 'y' para eliminar o 'n' para regresar al menú
y

==================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
==================================
e

1: Juan Arias
Revolución 22
Minatitlán, Ver. cp. 94233
juan@gmail.com
922-8024552

2: José Camacho
Independencia 142
Xalapa, Ver. cp. 96400
josec@gmail.com
928-4236652

3: Maria Flores
Roble 18
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

4: Norma Perez
Juan Escutia 1022
Minatitlán, Ver. 95000
norma@gmail.com
922-66665555

5: Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222

==================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
==================================
a

Ingresa el nombre del archivo:
```

```
datos.txt  <-- NOTA QUE ES EL MISMO archivo

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
e

1: Juan Arias
Revolución 22
Minatitlán, Ver. cp. 94233
juan@gmail.com
922-8024552

2: José Camacho
Independencia 142
Xalapa, Ver. cp. 96400
josec@gmail.com
928-4236652

3: Maria Flores
Roble 18
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

4: Norma Perez
Juan Escutia 1022
Minatitlán, Ver. 95000
norma@gmail.com
922-66665555

5: Manuel Soto    <-- ¿POR QUÉ sólo agregó este contacto y no se duplicó?
// SUGERENCIA: la clase de estructura de datos utilizada para
AddressBook.addressEntrryList no acepta duplicados.
¿Qué decidirás hacer?
Allende 100
Coatzacoalcos, Ver. 96400
manuel@gmail.com
921-4444333

6: Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
```

```
e) Mostrar
f) Salir
====================================
b

Nombre:
Pedro
Apellido:
Zarate
Calle:
Madero 218
Ciudad:
Minatitlán
Estado:
Veracruz
CP:
95000
Email:
pedro@gmail.com
Teléfono:
922-9999999

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
e

1: Juan Arias
Revolución 22
Minatitlán, Ver. cp. 94233
juan@gmail.com
922-8024552

2: José Camacho
Independencia 142
Xalapa, Ver. cp. 96400
josec@gmail.com
928-4236652

3: Maria Flores
Roble 18
Coatzacoalcos, Ver. 96535
maria@gmail.com
921-4233245

4: Norma Perez
Juan Escutia 1022
Minatitlán, Ver. 95000
norma@gmail.com
922-66665555

5: Manuel Soto
```

```
Allende 100
Coatzacoalcos, Ver. 96400
manuel@gmail.com
921-4444333

6: Pedro Zarate
Madero 218
MMinatitlán, Ver. 95000
pedro@gmail.com
922-9999999

7: Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222


====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
d

Ingrese apellido completo o primeras:
Za

Los siguientes 2 contactos se encontraron:
1: Pedro Zarate
Madero 218
MMinatitlán, Ver. 95000
pedro@gmail.com
922-9999999

2: Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
c

Ingrese el apellido a eliminar:
Za
```

```
1: Pedro Zarate
Madero 218
MMinatitlán, Ver. 95000
pedro@gmail.com
922-9999999

2: Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222

2

Ingresa 'y' para eliminar o 'n' para regresar al menú
Roger Zavaleta
Roman Marin 102
Nanchital, Ver. 96300
roger@gmail.com
923-2222222

y

====================================
Elige una opción del menú
a) Cargar de archivo
b) Agregar
c) Eliminar
d) Buscar
e) Mostrar
f) Salir
====================================
f
```