|  |  |  |
| --- | --- | --- |
| **Nombre**: Salvador Alejandro Fernández Díaz | | **Matrícula**: al04513405 |
| **Nombre del curso:** Computación en Java | **Nombre del profesor**: Daniel Horacio Chávez Argott | |
| **Modulo 1** | Actividad 4 | |
| **Fecha**: 14/06/2023 | | |
| **Bibliografía**: | | |

**Instrucciones**  
Inicia un nuevo proyecto y realiza lo siguiente.

* 1. Escribe una clase llamada AddressBook, que representará la agenda telefónica.
  2. La clase AddressBook deberá tener un atributo de tipo HashMap que almacene los contactos, HashMap será un Map<String, String>, el primer atributo String utilizará el número telefónico como llave.
  3. En la clase AddressBook se implementarán los métodos para la lectura y escritura de los contactos en un archivo de texto.
     1. **load**: cargará los contactos del archivo.
     2. **save**: guardará los cambios en el archivo.
  4. Se implementarán tres métodos para modificar la información de los contactos.
     1. **list**: mostrar los contactos de la agenda.
     2. **create**: crear un nuevo contacto.
     3. **delete**: borrar un contacto.
  5. Los tres métodos anteriores deberán estar conectados a un menú interactivo que proporcionará al usuario de la aplicación una interfaz para realizar las tres acciones anteriores:
     1. **list**: iterar el HashMap de AddressBook y mostrar los contactos con el siguiente formato:

|  |
| --- |
| Contactos: {Número}: {Nombre} {Número} : {Nombre} {Número} : {Nombre} {Número} : {Nombre} .... |

* 1. **create**: solicitar el nombre y número de contacto a guardar.
  2. **delete**: solicitar el número telefónico a eliminar.

1. Crea un repositorio en línea y sube el proyecto.
2. Realiza un reporte sobre la solución implementada para el problema, explicando el funcionamiento del programa.
   1. Incluye capturas de pantalla como evidencia del funcionamiento del programa.
   2. Incluye la liga al repositorio en el reporte realizado.
3. Eres libre de incluir funcionalidades adicionales que pueden tomarse en cuenta como puntos extra.

Nota: El archivo será un archivo de texto plano que almacenará los contactos en un formato de CSV (Comma Separated Values).

|  |
| --- |
| {Número},{Nombre} {Número},{Nombre} {Número},{Nombre} {Número},{Nombre} |

**Entregable**

     Documento en formato DOC, DOCX o PDF, que incluye:

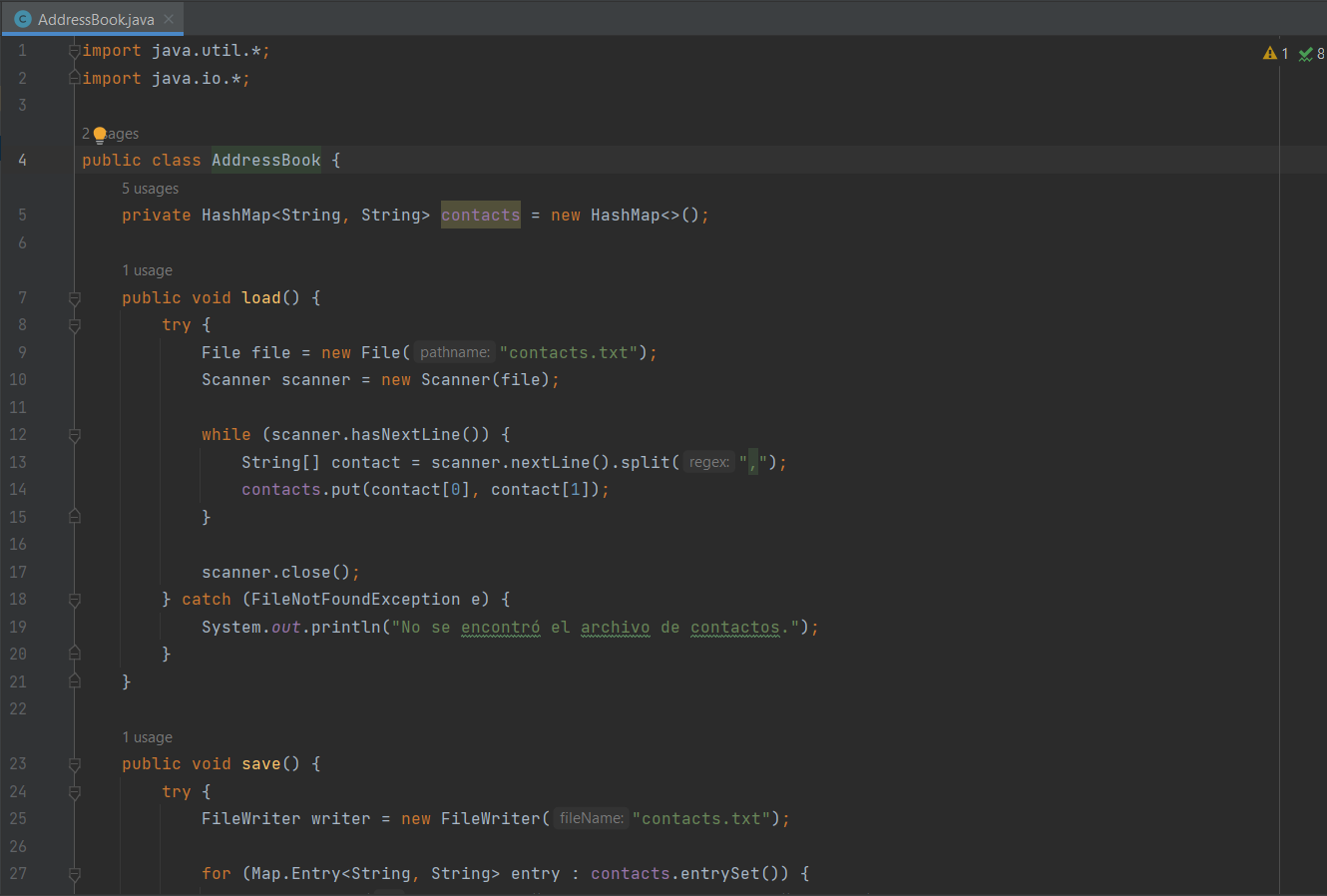
* + Reporte sobre la solución implementada para el problema, explicando el funcionamiento del programa.
  + Capturas de pantalla como evidencia del funcionamiento del programa.
  + Liga al repositorio en el reporte realizado.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

***Implementación de una Agenda Telefónica en Java***

**Introducción**

En este reporte, se presenta la implementación de una agenda telefónica en Java. El programa permite al usuario interactuar con una agenda telefónica, incluyendo la capacidad de listar contactos, crear nuevos contactos, eliminar contactos y guardar los cambios en un archivo.

**Descripción del Programa**

El programa consta de una clase principal: **AddressBook**. Esta clase representa la agenda telefónica y contiene un atributo **contacts** que es un HashMap. El HashMap utiliza el número de teléfono como clave y el nombre del contacto como valor.

La clase **AddressBook** proporciona varios métodos para interactuar con la agenda:

* **load()**: Carga los contactos desde un archivo de texto.
* **save()**: Guarda los contactos en un archivo de texto.
* **list()**: Muestra los contactos en la agenda.
* **create(String number, String name)**: Crea un nuevo contacto.
* **delete(String number)**: Elimina un contacto.

El programa también incluye un menú interactivo que permite al usuario seleccionar qué operación quieren realizar.

**Funcionamiento del Programa**

Cuando el programa se inicia, primero carga los contactos del archivo de texto. Luego, muestra un menú interactivo que permite al usuario elegir entre listar los contactos, crear un nuevo contacto, eliminar un contacto o guardar los contactos. El menú se repite hasta que el usuario elige salir del programa. Cuando el usuario elige guardar los contactos, el programa guarda los contactos actuales en el archivo de texto.

**Liga al repositorio:** https://github.com/Chavafd/Actividad-4.git