

Enunciado del Proyecto

Introducción

Se desea realizar una aplicación para manejar una agenda de contactos. La implementación de la misma se deberá realizar utilizando la clase ArrayList. Establecer un límite de contactos en la agenda, y esta limitación tiene que ser tenido en cuenta durante el desarrollo de la aplicación.

Tenemos dos tipos de contactos: familiares y amigos. Para cada uno de los contactos se deberá almacenar obligatoriamente su nombre, apellidos, dirección de correo electrónico y teléfono. Además, de los familiares almacenaremos la dirección de su vivienda y la relación de parentesco que tenemos con ellos, que podrá ser: cercana o lejana. Para los amigos almacenaremos también el grado de afinidad que tenemos con ellos (un entero entre 1 y 5, siendo 1 el grado de menor afinidad y 5 el de mayor afinidad).

Cada clase debe tener obligatoriamente: constructores por defecto y con parámetros, métodos getter y setter por cada uno de los atributos y un método toString. Además, deberás añadir todos los métodos que consideres necesarios, para que la aplicación funcione correcta y eficientemente.

Funcionamiento de la aplicación

Al abrir la aplicación el usuario se encuentra con el siguiente menú:

1. Insertar contacto.
2. Eliminar contacto.
3. Modificar contacto
4. Mostrar contacto.
5. Mostrar todos los contactos.
6. Mostrar todos los contactos de un tipo.
7. Guardar agenda
8. Cargar agenda.
0. Salir.

Insertar contacto

Al seleccionar insertar un contacto hay que tener en cuenta que no puede haber dos contactos con el mismo teléfono, independientemente del tipo de contacto que sean.

Al seleccionar esta opción nos deberá aparecer otro menú con las siguientes opciones:

1. Familiar.
2. Amigo.
0. Volver al menú principal.

Y si selecciona familiar o amigo deberemos tratar de insertar un contacto del tipo seleccionado.

Eliminar contacto

Solicita el teléfono y borra el contacto cuyo teléfono sea introducido, siempre que esté en la agenda. No distingue entre tipo de contacto.

1. Modificar contacto

Al seleccionar esta opción nos deberá aparecer otro menú con las siguientes opciones:

- | |
|---|
| <ul style="list-style-type: none">1.- Modificar Teléfono2.- Modificar Correo electrónico.3.- Modificar Dirección0.- Volver al menú principal |
|---|

Mostrar contacto

Dado un teléfono muestra los datos del contacto, si existe uno con ese teléfono. No distingue entre tipo de contacto.

Mostrar todos los contactos

Debe mostrar todos los contactos de nuestra agenda ordenados en orden creciente por número de teléfono. Al finalizar el listado deberá aparecer el número total de contactos almacenados en nuestra agenda.

Mostrar todos los contactos de un tipo

Al seleccionar esta opción nos deberá aparecer otro menú con las siguientes opciones:

- | |
|--|
| <ul style="list-style-type: none">1. Familiares.2. Amigos.0. Volver al menú principal. |
|--|

Si seleccionamos familiares, el programa muestra un listado de todos los familiares almacenados en nuestra agenda. Al finalizar el listado deberá aparecer el número total de familiares almacenados en nuestra agenda.

Si seleccionamos amigos, el programa muestra un listado de todos los amigos almacenados en nuestra agenda, ordenados de forma creciente por grado de afinidad. Al finalizar el listado deberá aparecer el número total de amigos almacenados en nuestra agenda.

Toda la aplicación debe guiarnos mediante mensajes. Por ejemplo, nos tiene que indicar si el contacto ha sido insertado correctamente o no, si al listar contactos no existe ninguno, etc.

Guardar agenda

La agenda se guardará en un fichero, si el fichero no existe se creará

Cargar agenda

Se introducirá el nombre del fichero y se mostrará la agenda almacenada en dicho fichero

Requisitos

- Es obligatorio que exista una clase para los menús, no se debe hacer en el programa principal.
- Es obligatorio controlar si el usuario introduce una opción incorrecta, si fuese necesario mediante excepciones.
- Es obligatorio comentar toda la aplicación.

Calificación de la práctica. 5 puntos de evaluación

- Correcto funcionamiento de la aplicación (60%)

- Estructura de clases y métodos, reutilización de código, eficiencia de los algoritmos, etc. (40%)
- La no presentación de la práctica en plazo supondrá un 0 en su calificación.
- La copia de la práctica supondrá un 0 en su calificación para todos los alumn@s implicados.

Entrega

- **Fecha límite de entrega:** Lunes 24 de Mayo a las 23:55.
- **Presentación:** 25,26,27 de Mayo del 2021