

Tarea 1 – ArrayList

Enunciado

1. Crea un programa que implemente un teléfono móvil y sus contactos con las siguientes capacidades.

- a. Implementa la clase `Contacto` con los siguientes atributos:
 - i. Dos campos, ambos String, uno llamado `name` y el otro `phoneNumber`.
 - ii. Un constructor con dos Strings.
 - iii. Y tres métodos: `getName`, `getPhoneNumber` y `createContact` que recibe dos parámetros de tipo String (el nombre de la persona y el número de teléfono) y devuelve una instancia de `Contacto`. Este es el único método que es estático.
- b. Implementa la clase `TelefonoMovil` con los siguientes atributos y métodos:
 - i. Dos campos, un String llamado `myNumber` y un `ArrayList` de tipo `Contacto` llamado `myContacts`.
 - ii. Un constructor que recibe el número de teléfono e instancia `myContacts`.
 - iii. `addNewContact`, recibe un parámetro de tipo `Contacto` y devuelve un valor booleano. Devuelve verdadero si el contacto no existe y se ha podido añadir a la lista o falso si por el contrario el contacto ya existe (no debemos agregar el mismo contacto dos veces, con el mismo nombre).
 - iv. `updateContact`, recibe dos parámetros de tipo `Contacto` (el contacto antiguo que se actualizará con el nuevo contacto) y devuelve un valor booleano. Devuelve verdadero si el contacto existe y se actualizó correctamente o falso si el contacto no existe. Ten en cuenta, que no se puede actualizar con un contacto que ya exista en la lista de contactos.
 - v. `removeContact`, recibe un parámetro de tipo `Contacto` y devuelve un valor booleano. Devuelve verdadero si el contacto existe y se eliminó con éxito, o falso si el contacto no existe.
 - vi. `findContact`, recibe un parámetro de tipo `Contacto` y devuelve un int. El valor devuelto es su posición en `ArrayList`, será -1 (no existe) o un valor mayor o igual a 0 (sí existe).
 - vii. `findContact`, igual que arriba, solo que recibe un parámetro de tipo `String`. Busca en la lista si existe un contacto con ese nombre.
 - viii. `queryContact`, recibe un parámetro de tipo `String` y devuelve un contacto. Usa el `String` para buscar el nombre y luego devuelve el contacto. Devuelve nulo en caso contrario.
 - ix. `printContacts`, no tiene parámetros y no devuelve nada. Imprime los contactos en el siguiente formato:

Lista de contactos:

1. Pepe → 31415926
2. Alicia → 16180339
3. Tomás → 11235813
4. Jessica → 23571113

- c. Crea una clase `Main` con un método `main` y agrega un menú de opciones como vimos en la teoría. El menú debe mostrar las siguientes opciones en ese orden:

- i. 0 salir.
- ii. 1 imprimir contactos, llamará a `printContacts`.

- iii. 2 agregar un nuevo contacto, pedirá al usuario el nombre y número de teléfono, creará el contacto e intentará añadirlo a la lista, si todo va bien mostrará un mensaje de éxito y si no se ha podido añadir mostrará un mensaje de error.
- iv. 3 actualizar un contacto existente, primero preguntará el nombre del contacto y comprobará que existe con `queryContact`, después pedirá el nuevo nombre y el nuevo número de teléfono, creará un nuevo contacto y llamará a `updateContact`. Si todo va bien imprimirá un mensaje de éxito y sino de error.
- v. 4 eliminar contacto de la lista, pedirá el nombre, buscará al contacto en la lista y lo eliminará. Informará al usuario.
- vi. 5 buscar/encontrar contacto por nombre, pedirá el nombre, buscará el contacto e informará al usuario con la información del contacto.
- vii. 6 volver a imprimir la lista de opciones.

NOTA: En `TelefonoMovil`, usa el método `findContact` en los otros métodos (excepto `printContacts`) para verificar si existe el contacto antes de continuar.

NOTA: Dos objetos de contacto son iguales si tienen el mismo nombre.

NOTA: Todos los métodos deben definirse como públicos (excepto los dos métodos `findContact()` que son privados).

NOTA: El método `indexOf` de `ArrayList`, internamente usa el método `equals` que heredamos desde la clase `Object`. Si no lo sobreescrivimos en nuestra clase, utiliza el id del objeto para comprobar si dos objetos son iguales. Pero en nuestro caso dos objetos de tipo Contacto son iguales si tienen el mismo nombre y mismo número de teléfono.

Entrega

- Sube un PDF con las respuestas y el código copiado.