FUNDACIÓN UNIVERSITARIA DE POPAYÁN (FUP)

INTEGRANTES:

MANUEL ESCOBAR FIGUEROA
SEBASTIAN MAMIAN PALTA

DIRIGIDO A:

ADRIAN DANILO ASTUDILLO

19 DE NOVIEMBRE DEL 2024
FACULTAD DE INGENIERÍA
INGENIERIA DE SISTEMAS
POPAYÁN - CAUCA

1. ¿Qué ventajas nos ofrece la POO en este ejercicio comparado con una programación estructurada?

La Programación Orientada a Objetos (POO) hace que el código sea mucho más organizado y manejable, especialmente cuando el proyecto crece o se vuelve más complejo. En lugar de escribir funciones que interactúan con datos de manera global, la POO nos permite crear "bloques" de código llamados objetos, que se encargan de tareas específicas.
la clase Person es un objeto que tiene todos los detalles de una persona (como su nombre, ciudad, etc.), y la clase Song se encarga de manejar las canciones. Cada objeto tiene sus propios datos y métodos, lo que facilita el mantenimiento y actualización del código sin afectar el resto del programa. En programación estructurada.
Además, la POO permite reutilizar el código. Por ejemplo,crear muchas instancias de Person o Song sin tener que repetir todo el código cada vez. En un enfoque estructurado, que copiar y pegar código, lo cual puede ser muy tedioso y propenso a errores.

2. ¿Qué problemas se podrían presentar si no se utiliza el encapsulamiento en los datos de cada persona?

El encapsulamiento es como una "caja de seguridad" que protege los datos del programa. Si se usa encapsulamiento, cualquiera podría modificar directamente la información de una persona o una canción, sin ningún control. Esto puede provocar muchos problemas:
Acceso sin restricciones: Imagina que alguien, en cualquier parte del programa, cambia el nombre de la persona a algo incorrecto, o borra su ciudad de residencia sin querer. Sin encapsulamiento, el programa no tiene forma de impedir que esto pase.
Error humano: Si un dato importante como el email no está validado antes de ser modificado, el usuario podría escribir algo como "miemail.com", lo que no tiene sentido, pero no habría manera de detenerlo.
Confusión a largo plazo: Si en el futuro decido cambiar la estructura interna de la clase Person (por ejemplo, cambiar el tipo de datos de un atributo), tendría que ir cambiando todo el código que accede a esos atributos. Con encapsulamiento, podria modificar la forma en que se almacenan los datos sin que eso afecte a las demás partes del programa.
El encapsulamiento protege y controla cómo se accede y modifica la información, lo que hace que el programa sea más seguro y fácil de mantener.

3. ¿Cómo podría mejorarse este programa si se quisiera agregar más de tres canciones por persona?

Actualmente, el programa solo permite agregar tres canciones por persona, porque está limitado por el tamaño de un arreglo fijo (SIZE = 3). Si quiero que una persona pueda tener más canciones, hay un par de formas de hacerlo de manera más flexible:
Usar listas dinámicas: En lugar de usar un arreglo con un número fijo de elementos, podría usar una lista (ArrayList). Las listas permiten agregar elementos de forma ilimitada hasta que el sistema se quede sin memoria, sin tener que preocuparte de definir un tamaño.
Por ejemplo, puedo cambiar la línea donde declaro las canciones en la clase Person para usar un ArrayList: