IMPLEMENTE EL CÓDIGO FUENTE EN EL LENGUAJE DE PROGRAMACIÓN DE SU PREFERENCIA PARA RESOLVER LOS SIGUIENTES EJERCICIOS

La implementación debe ser hecha sin contemplar solicitud de datos al usuario ni interfaz gráfica, si no, con datos ficticios de prueba, escritos desde los métodos de inicio de los programas. Se deben hacer las impresiones del caso para demostrar la funcionalidad y correctitud de los programas.

1. Realizar la implementación de un programa que emule un editor gráfico de documentos que admita el agrupamiento, lo cual es un concepto que se utiliza en toda una gama de editores gráficos. Suponer que el documento consta de varias hojas. Cada hoja contiene objetos representables, incluyendo textos, objetos geométricos y grupos. Un grupo es, simplemente, un conjunto de objetos representables, inclusive otros grupos de manera “recursiva”. Los objetos geométricos pueden ser círculos, elipses, rectángulos, líneas y cuadrados.

Cree instancias para demostrar el uso exhaustivo de cada uno de los elementos del editor gráfico de documentos. Utilice ocultamiento de información y declare solo los get/set necesarios para su implementación

1. Implementar una versión muy simplificada de un sistema para una biblioteca. En el sistema aparecen socios, que se registran en la biblioteca y a partir de ese momento pueden tomar prestados libros de la misma.

Un socio está caracterizado por un número de socio, un nombre y una dirección; además, en cada momento se puede saber el número de libros que un socio tiene prestados. Por su parte, de cada libro se conoce su código, título, autor y si está o no disponible; además se puede saber en cualquier momento la localización del libro en la biblioteca (si está disponible o si lo tiene algún socio).

Los libros se prestan a los socios, y como consecuencia aparece la noción de préstamo; un préstamo estará caracterizado, además de por el código del libro prestado y el número de socio, por la fecha del mismo. Por otra parte, el sistema también permite llevar control de los socios que tengan prestados más de 3 libros (usar filter para esta acción).

Demuestre el funcionamiento de su código a partir de escenarios puntuales con valores ficticios pero que permitan ejecutar las acciones principales de las clases. Agregue datos, cree socios, haga que reserven libros, muestre estados de libros, muestre socios que tengan prestados más de 3 libros, etc.

1. A partir de la implementación del problema hecho en clase sobre agendas de contactos y eventos, modifique el código para agregar dos patrones de diseño: Singleton y Abstract Factory al problema. Específicamente para el patrón Singleton, documente en el código las diferencias entre un Eager Singleton y un Lazy Singleton y defina cual le parece que sea mejor utilizar en la implementación del problema y por qué?

Ref Example: <https://www.linkedin.com/pulse/two-types-singleton-design-pattern-lazy-eager-arifuzzaman-tanin/>