



Patrones de diseño de Creación

Ejercicio 1

Deberá crear un programa que se encargue de manejar una matriz de empleados (con nombre, apellido y cédula). Deberá tener los métodos para agregar un elemento a una determinada fila y columna, otro para agregar un elemento al final de una fila y otro para copiar una fila y agregarla al final de la matriz. Debe tomar en cuenta que los elementos de la fila copiada deben ser diferentes a los elementos de la fila original. Además, tome en cuenta que no se podrá manejar más de una matriz en el programa.

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa el funcionamiento de los métodos y la utilización correcta del patrón.

Ejercicio 2

Suponga que una empresa de desarrollo de software requiere una aplicación que maneje los proyectos que tiene a su cargo. Para un proyecto se necesita manejar la siguiente información: un nombre (texto), un jefe de proyecto (con nombre, apellido y cédula), un tiempo en semanas (entero) y una lista de fechas en las cuales se deben realizar adelantos del proyecto. En el caso de que para el proyecto no se deban realizar adelantos, la fecha en la lista será la fecha de entrega final del proyecto.

Deberá analizar e implementar todas las validaciones correspondientes a los diferentes atributos del proyecto para que este se cree correctamente.

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa la creación de proyectos y la utilización correcta del patrón.

Ejercicio 3

Modifique el ejemplo de Abstract Factory del libro de patrones (Applied Java Patterns), en el que crean direcciones, para que sea dos problemas diferentes de Factory Method, o sea 2 programas diferentes. Tome en cuenta que para lograrlo tiene que partir el contexto del problema para tener familias de clases más pequeñas. Debe dividir el código en 2 programas independientes incluyendo la prueba del programa.

Ejercicio 4

Deberá crear un programa que se encarga de gestionar la información de venta de seguros para autos, para hogares y personales. Para cada uno de ellos hay diferentes paquetes según su cobertura: básico, estándar y completo. Debe determinar al menos 3 posibles atributos en cada clase para lo cual debe analizar cuáles clases pueden compartir datos. Para cada seguro deberá implementar un método para la descripción del seguro, por ejemplo "Seguro: Hogar, Cobertura: básico, cédula propietario: 2044....".

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa la creación de los diferentes seguros y la utilización correcta del patrón.

Ejercicio 5

Cree una aplicación que se encargue de administrar los montos a cobrar por la publicidad en los periódicos del Grupo Nación. El grupo nación tiene a su cargo la creación de 3 periódicos diariamente: La Nación, Al día y La Teja. Para el cobro de publicidad se tiene que almacenar el mensaje a mostrar en la publicidad. Para el cobro se debe tomar en cuenta si la publicidad es de un cuarto de página, de media página o de página completa). Dependiendo del periódico se cobra diferente el tipo de publicidad



(montos constantes diferentes para cada periódico). Debe tomar en cuenta que La Teja no maneja publicidad de página completa y La Nación no maneja publicidad de un cuarto de página.

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa la creación de los diferentes tipos de publicidad y mostrar el mensaje, el tipo, el periódico y el monto de la publicidad. Además, de la utilización correcta del patrón.

Ejercicio 6

Debe crear una clase ventana que simule una ventana rudimentaria de chat, en la cual se tenga un JTextArea para ver el texto de la conversación, un JTextField para ingresar nuevo texto, un botón para agregar el texto del JTextField a la conversación y uno para refrescar el texto(dado que es rudimentaria no se visualiza en tiempo real si la otra persona escribe varias veces y yo no he escrito nada). La conversación debe compartirse entre las diferentes ventanas.

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa el programa con al menos 2 ventanas y la utilización correcta del patrón.

Ejercicio 7

Cree un programa que se encargue de manejar los profesores y los estudiantes de una universidad(al menos 3 atributos básicos cada uno). Deberá manejar profesores interinos(manegan fecha de inicio y fecha fin de nombramiento, tiempo que labora en cuartos (1, 0.75, 0.5, 0.25 – el primero sería tiempo completo y el último un cuarto de tiempo) y salario general que reciben los interinos. Deberá manejar profesores en propiedad (tienen fecha de nombramiento, tiempo que labora y pago general que reciben los profesores en propiedad). Deberá manejar estudiantes becados(número de beca, y pago general por cada tipo de beca) y los estudiantes por convenio(tienen la institución de dónde proviene y una lista de cursos que le fueron convalidados, el caso de que se le hayan convalidado cursos).

Deberá analizar e implementar todas las validaciones correspondientes a los diferentes atributos de los diferentes tipos de elementos de la universidad para que este se cree correctamente.

Debe tomar en cuenta que, aunque no hay implementación gráfica, la elección del elemento a crear se realizaría en la interfaz gráfica.

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa la creación de todos los diferentes tipos de participantes y la utilización correcta del patrón.

Ejercicio 8

Realice un programa que simule el almacenamiento de los movimientos de un juego de ajedrez. Se debe almacenar en una lista los diferentes movimientos del juego. Cada movimiento tiene una letra correspondiente a la fila, un número correspondiente a la columna, una letra correspondiente a la pieza y el bando que realizó el movimiento(blanco o negro). Este juego en particular tiene la posibilidad de almacenar un estado del juego al cual se puede regresar si el jugador lo desea, por lo cual debe manejar un método que se encargue de almacenar el posible estado y otro para que se encargue de devolver el juego a ese estado. Para el ejercicio sólo deberá crear la clase que almacena los movimientos y los métodos mencionados, y no la interfaz gráfica del mismo.

Debe analizar cuál o cuáles son los patrones más adecuados para resolver el problema.

Debe crear una clase que compruebe de forma completa el funcionamiento de los métodos y la utilización correcta del patrón.