

Trabajo Práctico No. 1: Patrones de Diseño

Diseñar e implementar en Java (cuando se indique) cada uno de los siguientes problemas. Identifique los patrones de diseño aplicados en la solución y analice el resultado de su aplicación.

- 1) En un congreso de Aritmética decidieron utilizar un sistema para organizar sus expresiones matemáticas. Las expresiones con las que trabajan son expresiones básicas, que incluyen suma, resta, producto, división, potencia y raíz. Por ejemplo:
 - $(2+3)*5$
 - $(5+5)^{(3-1)}$
 - $((7+8) / (5-3))$

Luego de muchas discusiones se pusieron de acuerdo en qué es lo que querían que el sistema fuera capaz de proveer:

- a) Obtener el valor de la expresión: en los ejemplos anteriores, serían 25; 100; 7,5 respectivamente
- b) Obtener una lista con sólo los números de la expresión: en los ejemplos serían [2,3,5]; [5,5,3,1]; [7,8,5,3]
- c) Obtener una lista con sólo los operadores de la expresión: en los ejemplos serían [+,*]; [+pot,-]; [+/,,-]
- d) Imprimir fácilmente las expresiones, por ello quieren que poder obtener un String con la expresión propiamente dicha incluyendo los paréntesis “ $(2+3)*5$ ”; “ $(5+5)(3-1)$ ”; “ $((7+8) / (5-3))$ ”.

Adicionalmente, el sistema debe ser capaz de retornar el cálculo opuesto de una expresión. Esto quiere decir, si la expresión es una suma, el cálculo opuesto hace la resta, si es un producto hace la división, si es una potencia hace la raíz cuadrada, y así con cada operador. Por ejemplo, en la primera expresión presentada el cálculo opuesto sería $((2-3)/5)$ es decir -0,2. En el segundo ejemplo, el cálculo opuesto sería $3+15-5$ es decir 0. En el tercer ejemplo, el cálculo opuesto sería $((7-8) * (5+3))$, es decir -8.

Los matemáticos desean realizar un mecanismo en el cual sea posible “normalizar” los valores de una expresión, es decir si tenemos la expresión $(4+5)$ y se la normaliza dividiendo con el valor 10 la expresión resultante sería $(0,4 + 0,5)$. Si se la normaliza con el valor 2, la expresión resultante sería $(2 + 2,5)$. Para realizar esta operación, se genera una expresión nueva y la original no se modifica. La expresión generada debe respetar la misma estructura de la expresión original. También es factible normalizar aplicando primero una corrección (suma o

resta) y luego una raíz cuadrada. Por ejemplo si la corrección es suma 2 y luego raíz cuadrada la expresión resultante sería $(2,449489743 + 2,645751311)$. Es factible además aplicar cualquier combinación de operaciones matemáticas.

Finalmente, se debe incorporar la capacidad de generar una copia de una expresión. Cuando se duplica una expresión se realiza una copia de todos los elementos contenidos en ella. Implementar la solución al problema del organizador de expresiones en JAVA, tener en cuenta todos los conceptos vistos en clase. Prestar especial atención a la abstracción de funcionalidad.

Realizar una implementación prototipo en Java.

- 2) *Candy*. Se desea implementar el control de tableros de un juego de Golosinas (similar al Candy Crush). Cada ficha del tablero posee una fortaleza (golpes necesarios para destruirla), ocupa un espacio del tablero (medido en cantidad de casilleros), y tiene un poder de destrucción. Por ejemplo el “chocolate” tiene una fortaleza de 1, ocupa 1 casillero del tablero y tiene poder de destrucción 0; la “piedra” tiene fortaleza 6, ocupa 1 casillero de tablero y tiene poder de destrucción 0; el “caramelo a rayas” ocupa 1 casillero del tablero, tiene fortaleza 1, y un poder de destrucción de 10; la “torta” ocupa 4 casilleros, tiene fortaleza 8 y poder de destrucción 4, y así con cada una de las distintas fichas. También existen fichas especiales cuyo poder de destrucción se calcula como la fortaleza de la ficha, dividido el espacio que ocupan.

Un tablero posee un puntaje mínimo que es necesario alcanzar para pasar al siguiente nivel, una lista de fichas, la dificultad del mismo (se calcula como la sumatoria de todas las fortalezas de las fichas, dividido la sumatoria del poder de destrucción de todas las fichas).

En cada tablero, se debe poder buscar:

- Todas las fichas que posean un poder de destrucción mayor a 2
- Todas las fichas que ocupen más de 4 lugares
- Todas las fichas que tengan una fortaleza mayor a 5
- Combinaciones lógicas de los anteriores

- 3) *Fútbol*. Un club de fútbol necesita un sistema que le permita llevar control de sus socios y alquileres de canchas. Cada socio posee un nombre y apellido, una edad, si tiene paga la última cuota y una lista que contiene información de los alquileres de canchas que ha realizado. Cuando un socio alquila una cancha, se registra la fecha, el ID de la cancha que alquiló y cuánto pagó por el alquiler.

Cuando el dueño del club arriba al predio, le pide a los que administran el sistema, diferentes listados, para comprobar cómo está funcionando el club:

- a) Todos los socios con cuota impaga, ordenados alfabéticamente (primero por apellido, y si hay varios con el mismo apellido por nombre)
- b) Todos los socios menores de edad, ordenados por edad.

- c) Todos los socios que alquilaron alguna vez la cancha N°13, ordenados cantidad total de alquileres de esa cancha.
 - d) Todos los socios que pagaron más de \$500 por algún alquiler de cancha, listando primero los que tienen las cuotas pagas, ordenados alfabéticamente y luego los morosos, también ordenados alfabéticamente.
 - e) Cualquier combinación lógica de búsquedas.
- 4) *Streaming*. Una plataforma de streaming de video permite a sus usuarios consultar su catálogo de películas disponibles. De cada película se guarda el título, la sinopsis, los géneros (“acción”, “terror”, “drama”, etc), el director, los actores, el año de estreno, la duración de la película (en minutos), y la edad mínima requerida para ver la película.

La plataforma debe permitir realizar búsquedas de películas según diversos criterios, por ejemplo:

- Buscar todas las películas que en el título contenga la palabra “luna”.
- Buscar todas las películas que contengan un género específico (por ejemplo, “terror”).
- Buscar todas las películas en las que haya actuado Will Smith y cuyo director NO haya sido Martin Scorsese.
- Buscar todas las películas que se hayan grabado antes del 2015 y cuya duración sea menor a 95 minutos.

A su vez, dado que algunas películas son más exitosas que otras, a la hora de analizar si adquirir la licencia de una nueva película, la plataforma analiza si la película resultará rentable o no. Una película será rentable si cumple con ciertos criterios que hace que los usuarios prefieran verla. Por ejemplo, el CEO de la plataforma observó que los usuarios suelen ver:

- Películas cuya duración sea menor a 120 minutos, que no sean del género “comedia”.
- Películas posteriores del 2017, a no ser que se trate de películas del género “infantil” o “documental”.

Dado que las preferencias de los clientes cambian con el tiempo, el CEO de la plataforma de streaming desea poder cambiar en tiempo de ejecución las políticas de adquisición de licencias de películas, en caso de ser necesario. Proveer un servicio que le permita a la plataforma determinar si una película resultará rentable o no.

Realizar una implementación prototipo en Java.

- 5) Un sistema de recuento automático de votos para una elección registra los votos recibidos por los diferentes candidatos. Los candidatos poseen un nombre, un partido político y una agrupación.

Cada votante registra su voto en una mesa. Cada mesa tiene asociado un conjunto de personas que pueden votar en la misma (padrón), identificadas únicamente por su DNI. Las

mesas se encuentran distribuidas en diferentes establecimientos educativos, que a su vez se agrupan por barrio y localidad. Adicionalmente, puede haber mesas especiales que no pertenezcan a un establecimiento educativo sino al barrio o localidad directamente.

Cuando un votante se presenta a emitir su voto, se debe controlar que el mismo se encuentre en el padrón correspondiente. Si se cumple esta condición, se registra el voto de la persona, que puede contener una referencia a un candidato o ser un voto en blanco (sin referencia a ningún candidato). Adicionalmente el voto guarda la hora en que se emitió el mismo (el momento en el que se crea el voto).

Nota: Todos los votos emitidos se guardan en la mesa, aunque sean en blanco (es decir, siempre existe el voto, es lo mismo que sucede con un sobre en una elección real)

Implementar los siguientes servicios:

- a) Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer el porcentaje de votos recibidos por un candidato particular.
- b) Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer el porcentaje de votos en blanco recibidos.
- c) Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer el porcentaje de votos que se emitieron entre las 10:00 y las 11:00. (La clase `LocalDate` implementa la interface `Comparable`)
- d) Dada una mesa, un establecimiento educativo, un barrio o una localidad, conocer la cantidad total de votos que se emitieron.
- e) Obtener un listado de todos los candidatos que se presentan a la elección, ordenado por partido político si son del mismo partido ordenarlos por agrupación; y si son de la misma agrupación ordenarlos por nombre.

Aclaración: los porcentajes se calculan como la cantidad de votos que cumplen la condición dividido el total de votos emitidos en la mesa, establecimiento, barrio o localidad multiplicado por 10.

- 6) Una aseguradora provee distintos tipos de seguros dentro de su cartera, entre los mismos se encuentran:
 - **Seguro Automotor**, en el cual se cuenta con un número de póliza y una descripción (por ejemplo, “todo riesgo”, el monto asegurado, el DNI del dueño de la póliza).
 - **Seguro de Vida**, donde se registra el DNI del asegurado, el número de la póliza, su descripción y el monto asegurado.
 - **Seguro Hogar**, el número y descripción de la póliza, el monto del seguro y el DNI del dueño de la póliza.
 - **Seguro Integrador**, en el cual se guarda un cliente (solo el DNI) y los seguros con los que cuenta (Automotor, Hogar, Vida, Integrador, y cualquier otro tipo de seguro que ofrezca o pueda ofrecer la compañía). El monto asegurado de un seguro integrador se calcula como la suma de todos los seguros que contiene. El número de póliza de un seguro

integrador queda determinado por el mayor número de póliza de los seguros que contiene.

- **Seguro Temporal**, Está pensado para un seguro que tiene una fecha de inicio y fin, puede ser de cualquier tipo de seguro existente. El número de póliza es el mismo que la póliza contenida. El monto asegurado es el monto del seguro contenido siempre y cuando la fecha actual esté dentro del periodo de validez.

La Aseguradora posee diferentes formas de calcular el costo de la póliza. Por ejemplo, hay seguros automotores que se calculan como un monto fijo y otros que son un porcentaje del valor asegurado. En el caso del Seguro Hogar se calcula como un monto fijo más un porcentaje del valor asegurado. En el caso del seguro de vida es un porcentaje del valor asegurado. Sin embargo, la forma de calcular el costo del seguro puede cambiar en cualquier momento, e incluso se pueden incorporar nuevas formas de calcular el costo. Para el caso de seguro integrador es la suma de los costos de los seguros contenidos, en el seguro temporal el costo es el costo del seguro contenido, siempre y cuando la fecha actual esté dentro de la vigencia del seguro (caso contrario es 0).

Se desea brindar listados en los cuales se incluyan todos los seguros que posee la compañía que cumplan con determinado criterio. A continuación se listan algunos ejemplos.

- a) Listar todos los seguros cuya descripción contenga la palabra “Automotor”, ordenados por dni del asegurado, y luego por número de póliza si un asegurado posee más de un seguro que coincida con la búsqueda
- b) Listar todos los seguros cuya descripción contenga la palabra “Hogar” y el monto asegurado sea superior a \$250.000, ordenados por dni del asegurado
- c) Listar todos los seguros donde haya un DNI igual a 24.134.323, ordenado por número de póliza.
- d) Se pueden agregar nuevos tipos de búsquedas así como también combinaciones lógicas de las mismas. Los criterios de ordenamiento también pueden variar y deben poder combinarse de cualquier manera.