

Ejercitación 1 - Listener

Realizar los diagramas de clase correspondientes a cada ejercicio. Una vez modelado, comenzar a implementar el modelo en Java.

Ejercicio 1 - DigitalHospital (DH)

El hospital DigitalHospital quiere implementar un nuevo sistema para controlar a los pacientes que se encuentran internados allí. El hospital tiene muchos médicos especializados que se encargan de atender y controlar ciertos parámetros de cada paciente internado.

Un paciente está conectado a diferentes medidores digitales. En este momento, el hospital posee los siguientes medidores:

- Medidor de pulso.
- Medidor de oxígeno en sangre.
- Medidor de actividad cerebral.

Sin embargo en un futuro, el hospital desea incorporar más tecnología.

Cada médico tiene una especialidad diferente, por lo tanto, a cada médico le interesa suscribirse a diferentes medidores digitales. En el caso de que el paciente esté en alguna situación crítica, el medidor digital alertará a los médicos suscritos para que puedan actuar rápidamente.

Aclaración: Un medidor digital puede tener más de un médico suscrito. Un médico puede suscribirse a más de un medidor digital.

1. Crear el diagrama de clase correspondiente.
2. Implementar el modelo en Java

Ejercicio 2 - MONROE STREET BROTHERS

El **lobo** de la bolsa de **MONROE STREET BROTHERS** desea implementar un sistema para los **lobitos** que adquieren una determinada acción. La idea es que la acción que adquieren, les notifique los cambios de valores que se producen en ella. Además, en este mercado una acción puede ser comprada por varios lobitos.

La **acción** tendrá como responsabilidad, simplemente notificar a los lobitos que se encuentren suscritos a ella, sobre la suba de la acción y la baja de la acción, sin ninguna información adicional. En este primer modelo la acción no tiene ninguna otra propiedad que no sea la lista de suscriptores, tampoco tendrá otras responsabilidades que no sean las mencionadas.

Cada lobito tendrá como responsabilidad suscribirse a la acción de su interés.

1. Crear el diagrama de clase correspondiente.
2. Implementar el modelo en JAVA

Ejercicio 2 - Adicional:

Se agrega como información de la acción, el precio de la misma y dos responsabilidades:

- **aumentarValorDeAccion**, que recibe un porcentaje y aumenta el precio de la misma.
- **disminuirValorDeAccion**, que recibe un porcentaje y disminuye el precio de la misma.

Además, se modifican las notificaciones de suba y baja, enviando ahora la acción en cuestión. Por lo cual ahora los lobitos aparte de ser informados de que sube o baja alguna acción, ahora reciben en la misma la acción, para poder tomar mejores decisiones.

1. Modificar el diagrama de la clase anterior.
2. Implementar el modelo en JAVA

Ejercicio 3 - Digital Horse Racing Betting

El hipódromo quiere implementar un sistema para notificar el resultado de las **carreras** a todos aquellos interesados en los resultados. Sencillamente, cualquier interesado en el resultado se puede suscribir y será notificado. Al finalizar la carrera, el sistema notifica a todos los suscriptores el resultado de la misma.

Los interesados serán:

1. **Apostadores:** como este hipódromo es muy tramposo, los apostadores al ser notificados imprimirán por pantalla "Chorros, Chorros, Chorros"
2. **Kronika Tv:** este programa de periodismo amarillo imprimirá por pantalla "Estalló el verano en el hipódromo"
3. **Corredores de apuesta:** los corredores de apuestas tienen un patrimonio. Al ser notificados por el hipódromo los corredores aumentan su patrimonio en \$1000.

Ejercicio 3 - Adicional

En esta oportunidad, el hipódromo quiere incorporar un nuevo juego llamado "Quini 5". Este juego también notificará a todos sus suscriptores el resultado de los números.