Preguntas teóricas

1. Define hotspot y frozenspot
2. ¿En qué caso aplicar el template method en el diseño de un framework genera inversión de control?
3. ¿Qué es un patrón de diseño?
4. Los patrones de diseño se documentan siguiendo una estructura/ organización predefinida, ¿cuál es?
5. Describa y ejemplifique el patrón Composite
6. ¿Cuándo se aplica el patrón template method?
7. ¿Cuándo se aplica el patrón composite?
8. ¿Cuándo se aplica el patrón state?
9. ¿Cuándo se aplica el patrón adapter?
10. ¿Qué es un framework?
11. ¿Qué es hacer refactoring?
12. ¿Qué son los bad smells? Nombre 2 bad smells e indique, para cada uno, los refactorings que normalmente es necesario aplicar para resolverlos

Ejercicios

1. Sea un sistema de manejo de cuentas bancarias que maneja extracciones para varios tipos de cunetas. Para todas las cuentas una extracción es un proceso de varios pasos:
   1. Autorización: para una cuenta corriente, la extracción se autoriza sólo si el salgo de la misma no quedará por debajo del límite en descubierto (límite de crédito) de la misma. Para una caja de ahorro, la extracción se autoriza sólo si posee suficiente saldo (no hay extracciones en descubierto) y si no ha efectuado más de 4 extracciones en el mes corriente. Para tener una cuenta sueldo la extracción se autoriza sólo si posee suficiente saldo (no hay extracciones en descubierto)
   2. Extracción: para los tres tipos de cuneta, la extracción se efectúa restando el monto a extraer del saldo
   3. Cobrar costo: a las cuentas corrientes y las cajas de ahorro se les cobra el 12/1000 del valor extraído. Adicionalmente, a las cajas de ahorro se les cobra $2 por extracción. En las cuentas sueldos, no se cobran las extracciones (son gratis)
   4. Registrar movimiento. Para las cuentas, el movimiento se registra como una extracción (monto, fecha, etc.)
2. En un software para un sistema de alarmas, cada vez que se produce un evento distinto (por ejemplo que un sensor detecta un cambio), se envía un mensaje trigger a un objeto de la clase Alarma.

Dicho objeto reacciona al trigger de diferente manera si está activa, encendida o apagada.

Cuando está activa la reacción depende del abono que pagó el cliente de la alarma (en los otros casos no nos interesa):

* Para un abono normal, se hace sonar una bocina
* Para un abono premium, se llama a un teléfono proporcionado por el usuario
* Para un abono De Luxe, se habilita una cámara de seguridad
* Adicionalmente el usuario podría pedir que se agreguen nuevas opciones

Diseñe la clase Alarma y diga si es necesario usar algún patrón para atacar la posible complejidad de diseño (No es necesario codificar más allá de los encabezamientos de los métodos y algunas expresiones cuyo código es obvio, e.g bocina sonar

Use el pseudo lenguaje de su elección