

Simulador de cubiletes con dados configurables

Queremos desarrollar un simulador que permita representar distintos tipos de dados y cubiletes, para ser utilizados en juegos de mesa o de rol. El sistema debe permitir lanzar dados individuales, agruparlos en cubiletes, y obtener el resultado de un lanzamiento.

Cada dado puede ser lanzado individualmente, y devuelve un valor entero como resultado. Sin embargo, existen distintos comportamientos posibles al lanzarlo. Los comportamientos posibles son:

- Dado estándar: devuelve un número aleatorio entre 1 y 6.
- Dado con modificador fijo: es como el estándar, pero se le suma un valor fijo (por ejemplo, +2).
- Dado ponderado: tiene mayor probabilidad de devolver ciertos valores (por ejemplo, mayor probabilidad de sacar 6).
- Dado explosivo: si obtiene el valor máximo, se lanza de nuevo y se suma el resultado, repitiendo mientras siga saliendo el máximo.

Un cubilete contiene uno o más dados. Puede lanzarse para obtener un lanzamiento conjunto, que consiste en lanzar todos los dados que contiene y obtener un resultado global.

El resultado de un lanzamiento de cubilete debe permitir:

- Conocer los valores obtenidos por cada dado.
- Conocer el timestamp en el que se hizo el lanzamiento.
- Obtener el valor total (suma de todos los resultados).

Debe ser posible guardar el resultado del lanzamiento sin que posteriores lanzamientos del mismo cubilete lo afecten.

Provea:

1. Diagrama de clases UML.
2. Implementación en Java la funcionalidad requerida.
3. Pruebas automatizadas según las indicaciones dadas a continuación

Para las pruebas

Diseñe los casos de prueba teniendo en cuenta los conceptos de valores de borde y particiones de equivalentes vistos en la teoría. Documente los casos en una tabla como la que se muestra a continuación. Cada fila de la tabla corresponde a un método de una clase que usted cree que hay que testear. Por cada fila, indique con qué valores testear (el fixture).

Clase	Método	Fixture
ClaseEjemplo	metodoEjemplo(param)	Caso A variableEjemplo: 0 param: -1