

Pt4 Refactoring.

Opció 1: Glided Rose Kata.

Hola i benvinguts a l'equip Gilded Rose. Com ja sabeu, som una petita fonda amb una ubicació privilegiada en una ciutat destacada dirigida per un hostel simpàtic anomenat Allison. També comprem i venem només els millors productes. Malauradament, la qualitat dels nostres productes es deteriora constantment a mesura que s'acosten a la seva data de venda. Tenim un sistema que actualitza el nostre inventari per nosaltres. Va ser desenvolupat per un tipus anomenat Leeroy, que ha passat a noves aventures. La vostra tasca és afegir la nova funció al nostre sistema perquè puguem començar a vendre una nova categoria d'articles. Primer una introducció al nostre sistema:

- Tots els articles tenen un valor `SellIn` que indica el nombre de dies que tenim per vendre l'article
- Tots els articles tenen un valor `Quality` que denota el valor de l'article
- Al final de cada dia, el nostre sistema redueix els dos valors per a cada article

Bastant senzill, oi? Bé, aquí és on es posa interessant:

- Un cop superada la data de caducitat, la qualitat es degrada el doble de ràpid
- La qualitat d'un article mai és negativa
- "Aged Brie" augmenta realment en qualitat com més gran es fa
- La qualitat d'un article mai és superior a 50
- Les "sulfures", al ser un article llegendari, no s'han de vendre mai ni disminueixen la qualitat
- Els "passes de backstage", com el brie envellit, augmenta la qualitat a mesura que s'acosta el valor de `SellIn`; La qualitat augmenta en 2 quan hi ha 10 dies o menys i en 3 quan hi ha 5 dies o menys, però la qualitat baixa a 0 després del concert.

Heu de modificar el mètode `updateQuality` i afegir qualsevol codi sempre que tot funcioni correctament. Teniu un test per comprovar que a mesura que refactoritzeu el codi funciona correctament. No podeu modificar la classe `Item`.

Opció 2: Hero Kata.

Aquest repte està basat en el joc <https://en.wikipedia.org/wiki/HeroQuest>, però no cal saber de que va el joc per poder refactoritzar el codi.

El codi inicial té funcions amb molts paràmetres d'una banda, i la classe HeroQuest té una barreja d'atributs que no representen una única entitat.

En aquest repte, es pot modificar el test, i es poden crear/modificar les classes necessàries.