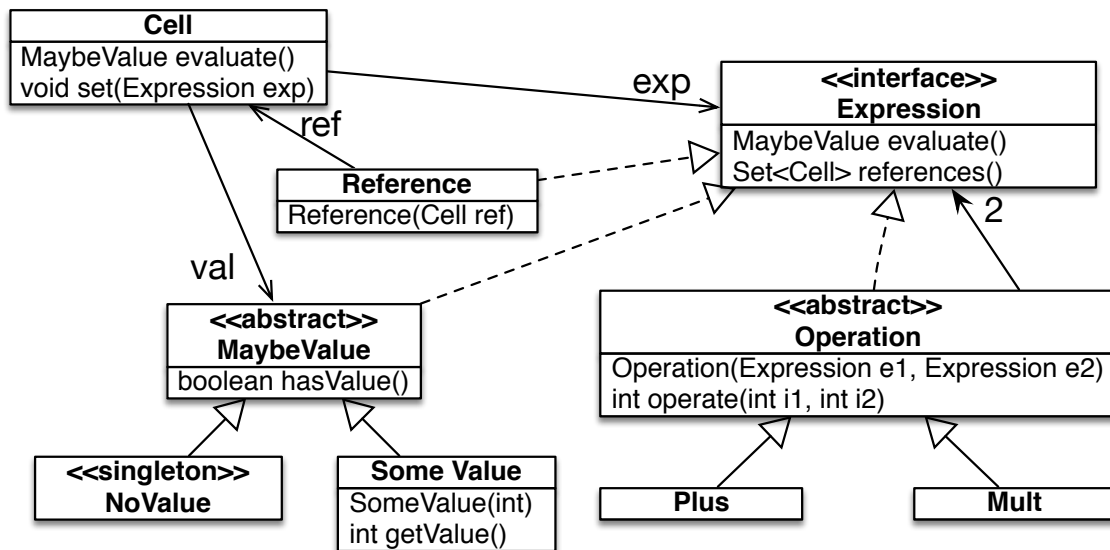


## Problema de disseny

Com a pas previ de la incorporació de la propagació de canvis a la pràctica del full de càlcul, heu arribat al següent disseny:



De moment, la implementació de la classe Cell és la següent:

```

1 public class Cell {
2     private Expression exp;
3     private MaybeValue val;
4
5     public Cell() {
6         this.exp = NoValue.INSTANCE;
7         this.val = exp.evaluate();
8     }
9
10    public void set(Expression exp) {
11        this.exp = exp;
12        this.val = exp.evaluate();
13    }
14
15    public MaybeValue evaluate() {
16        return this.val;
17    }
18 }

```

Totes les altres classes són immutables (és a dir, les instàncies no canvien una vegada creades).

El problema d'aquesta solució és que, com una expressió pot referenciar cel·les que han canviat de valor, el valor que una cel·la guarda pot quedar-se obsolet.

És per això que cal afegir un mecanisme que faci que, si alguna de les cel·les de les que depèn el valor de l'expressió assignada a una cel·la canvia de valor, aquesta cel·la recalcula automàticament el seu valor.

Per simplificar el problema, no cal que detecteu possibles cicles de dependències.

Es demana:

- Indiqueu quin patró fareu servir i perquè.
- Com es relacionen els elements que apareixen al patró amb aquest cas concret?
- Implementeu en Java la nova classe Cell (tant podeu fer servir la infraestructura de Java com no).