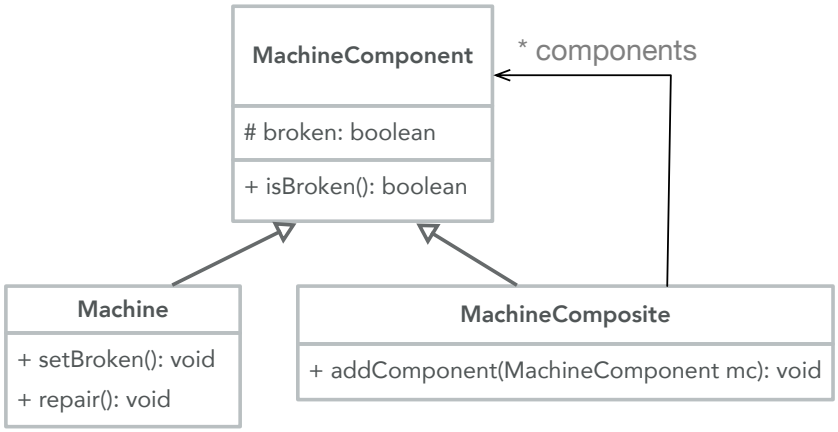
Comenteu mínimament el que preteneu fer i perquè, per a poder valorar millor la vostra solució. Sense una explicació de les decisions que preneu NO valoraré la solució.

Problema 1 (5 punts)

Donat el següent disseny de partida:



L'estat de trencament d'una màquina és diferent per cada subclasse:

- Una `Machine` està trencada o no trencada per ella mateixa (i, per això, tenim els dos mètodes por canviar el seu estat).
- En canvi, una `MachineComposite` està trencada si algun dels seus components està trencat.

Això seria trivial d'implementar de forma recursiva, però hi ha un problema: cada vegada que es vol saber l'estat de trencament d'un compost caldria travessar l'arbre de subcomponents i això té un cost.

Per això, algú ha pensat en el següent disseny: mantenir a cada component un booleà amb el seu estat, de manera que la implementació del mètode isBroken pot definir-se, **de forma única** a `MachineComponent`, com:

```
1. public class MachineComponent {
2.   protected boolean broken = false;
3.   public final[1] boolean isBroken() { return broken; }
4. }
```

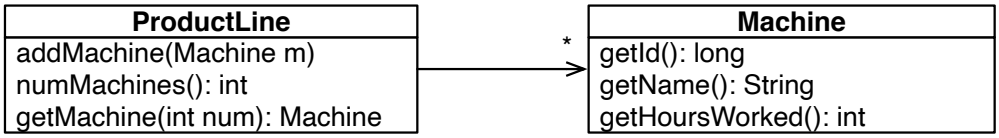
El que es demana és que completeu el disseny per a fer aquest disseny possible i **sense fer un recorregut de tots els subcomponents** per saber si una `MachineComposite` està trencada o no.

En aquest problema, per simplificar, **podeu suposar** que la jerarquia de components **no conté cap cicle**. Podeu modificar el disseny inicial, excepte la declaració de `broken` i la definició de `isBroken`.

[1] Un mètode final no pot redefinir-se a les subclases i, per tant, la seva implementación ha de ser vàlida per totes elles.

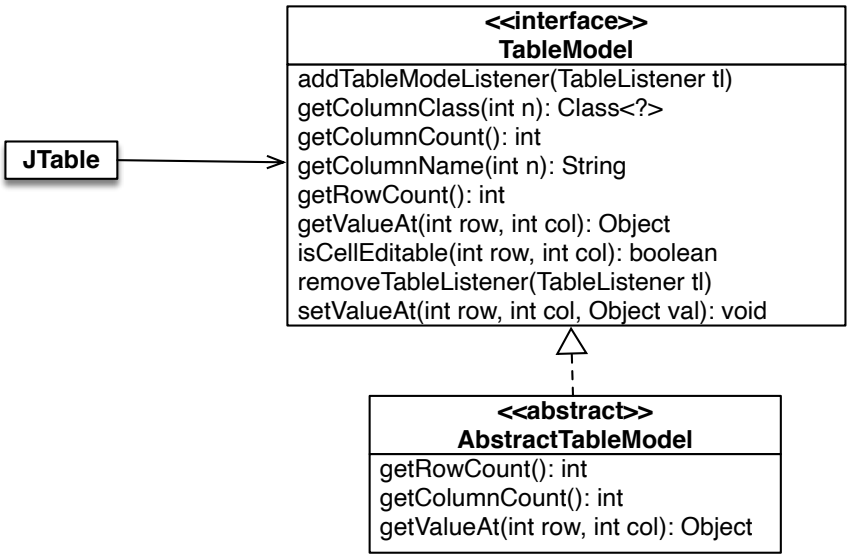
Problema 2 (5 punts)

Tenim les següents classes amb informació sobre les diferents màquines que formen part d'una línia de producció:



El que volem és mostrar, en forma de taula, la informació de les màquines d'una determinada línia de producció, de manera que, cada fila de la taula es correspondrà amb una màquina, i les columnes, es correspondran amb el `id`, `name` i `hoursWorked` de la màquina corresponent.

De cara a simplificar la programació usarem la classe abstracta `AbstractTableModel` que ens dóna una implementació parcial de les operacions de `TableModel`



En aquesta classe abstracta, els mètodes a implementar són els següents:

- getRowCount: retorna el nombre de files de la taula
- getColumnCount: retorna el nombre de columnes de la taula
- getValueAt: retorma el valor (com Object) a mostrar a la fila row, columna col (ambdues numerades des de 0).

El que es demana és:

a) Si fem que `ProductLine` estengui `AbstractTableModel`, quin(s) problema(es) de disseny estem creant?
b) Quin patró de disseny podem aplicar per a solucionar-lo(s)?
c) Què aconsegueix el patró que fa que se solucioni(n) el(s) problema(es)?
d) Implementeu les classes implicades en la vostra solució.