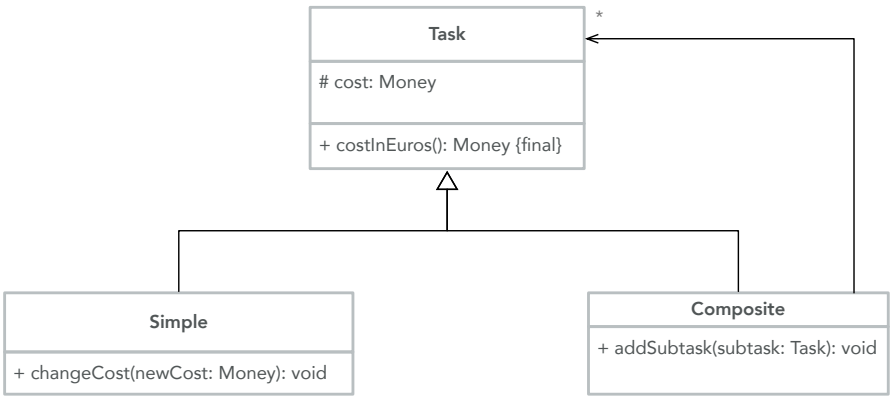
Donada la següent idea inicial de disseny:



En el que l'objectiu és aconseguir que tant les tasques simples com les complexes **mantinguin sempre actualitzar el valor de l'atribut protegit cost**, de manera que la implementació final de costInEuros sigui vàlida

```
public class Task {
  protected Money cost;
  protected Task(Money cost) {
    this.cost = cost;
  }
  public final Money costInEuros() {
    return this.cost;
  }
}
```

Les tasques Simples

- Tenen un constructor que rep una instància de Money amb el seu cost
- Tenen un mètode que permet canviar el seu cost

Les tasques Compostes

- Tenen un constructor sense paràmetres que les inicialitza
- Tenen un mètode que permet afegir una subtasca

El problema que presenta aquest disseny és el de mantenir actualitzat, en tot moment, el valor del cost ja que:

- Si una tasca simple canvia el seu cost, no tan sols canvia ella, sinó que han de canviar els costos de totes les tasques compostes que la contenen
- Si a una tasca composta li afegim una subtasca, no tan sols canviarà el seu cost, sinó que també canviarà el cost de totes les tasques compostes que la contenen.
- Si una tasca composta canvia de cost, no tan sols canviarà el seu cost, sinó que també canviarà el de totes les tasques compostes que la contenen.

El que heu de fer és **aplicar el patró observador** de manera que es pugui completar el disseny descrit. Això sí, **només notifiqueu canvis quan aquestos siguin reals** (és a dir, quan el cost realment canviï).

Per a fer-ho, de la biblioteca estàndar, podeu fer servir:

- Observador / Observable
- PropertyChangeSupport / PropertyChangeListener

Si utilitzeu Observador / Observable, és molt útil utilitzar el següent tipus per indicar les modificacions que s'han produït (model **push**):

```
public record CostChanged(Money oldCost, Money newCost) {}
```

Simplificacions addicionals:

- **Useu directament un int com a tipus Money i llenceu IllegalArgumentException si algú intenta crear una tasca amb cost negatiu** (en un disseny real definiríeu una classe immutable que utilitzés internament un BigDecimal amb el que controlar la precisió dels càlculs monetaris i, potser, un tipus per indicar divises)
- **Suposeu que la descomposició en subtasques forma un arbre**, per tant no cal que us protegiu davant de possibles cicles.

Instruccions per l'activitat:

- Resoleu individualment i a ma l'exercici (recordeu, un dels objectius de les activitats és que arribeu "entrenats" als exàmens) que entregareu escanejat.
- Poseu en comú una solució, a partir de les solucions individuals i, si voleu, havent programat i testejat la solució
- El codi que corregiré/puntuaré serà la solució consensuada (que ja no estarà a mà) i que entregareu amb un PDF amb els comentaris que creieu oportuns (**no vull un projecte sinó un PDF**).