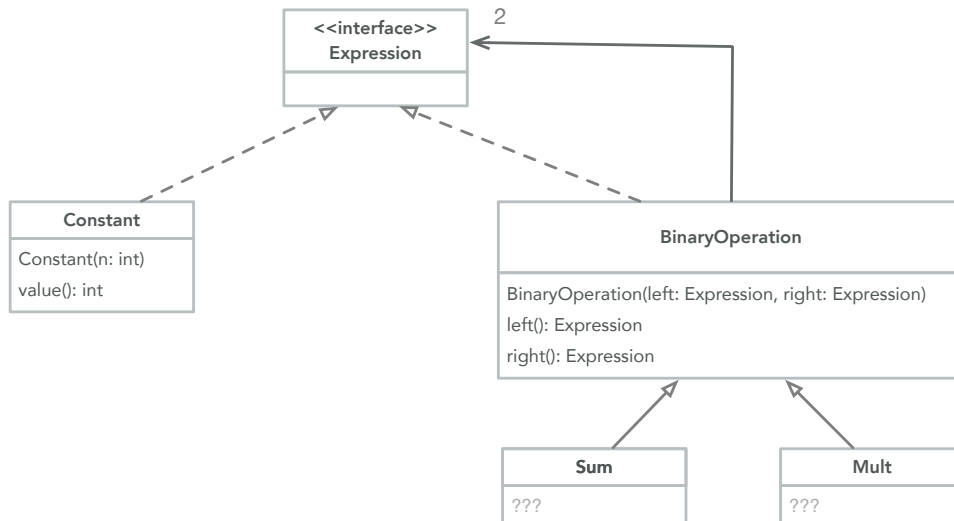


## Problema 1. (1 punt)

En Java modern, a partir de la versió 8, **què és un descriptor funcional?** Posa un exemple.

## Problema 2. (9 punts)

Donat el següent disseny inicial per a representar:



Aquest disseny permet construir **expressions aritmètiques** enteres i **immutables** formades per constants, sumes i productes. Per exemple, l'expressió **2 + (4 \* 8)** es construiria com:

```

Expression exp = new Sum(new Constant(2),
                        new Mult(new Constant(4),
                                new Constant(8)))
  
```

**A tots els apartats, expliqueu les decisions que preneu i el perquè del que feu (o no feu)**

- (1 punt) Implementeu** completament la **classe BinaryOperation** de manera que es comprovi en el seu constructor que cap de les subexpressions que es reben sigui null (llenant `NullPointerException`) i la **subclasse Sum**.
- (2 punts) Implementeu un mètode isValid()** per saber si una expressió és vàlida o no. Una expressió és vàlida si forma un arbre. Indiqueu (i feu) totes les **modificacions** que cal fer al disseny/codi existent.
- (1 punt) Esteneu el mètode value()**, que ara només s'aplica a les constants, **a totes les expressions**. Indiqueu (i feu) totes les **modificacions** que cal fer al disseny/codi existent i **perquè**. Podeu suposar, encara que no cal (**per què?**), que l'expressió és vàlida.
- (1 punt) Afegiu el que calgui per poder aplicar el patró visitant**, volent distingir entre `Constant`, `Sum` i `Mult`.
- (2 punts) Implementeu un visitant que determini si una expressió és vàlida**. Afegiu un **mètode estàtic** al visitant per iniciar el càlcul i definiu com a **privat** el seu constructor. No podeu usar la implementació de `isValid()` de l'apartat b.
- (2 punts) Implementeu un visitant que determini el valor d'una expressió**. Com abans, podeu suposar que l'expressió és vàlida. Afegiu un **mètode estàtic** al visitant per iniciar el càlcul i definiu com a **privat** el seu constructor. L'únic mètode `value()` que podeu cridar és el de la classe `Constant`.