

# Unit 3.2 Domain Layer Design

## Exercise 4 Solution

# Domain Layer Design – Exercise 4

**context** GimmeTheRate::newRate(cFrom: String, cTo: String, excRate: Float)

**pre** 1.1: cFrom <> cTo

**post** 2.1: enregistra que la proporció entre les divises és: cFrom = cTo\*excRate

**context** GimmeTheRate::convert(cFrom: String, cTo: String, amount: Integer): Float

**exc** 1.1 canvi-desconegut: no es disposa de conversió entre les divises

**post** 2.1: result = amount × (proporció entre les divises cFrom i cTo)

**context** CapaDomini::calculaPreu(v: Viatge, cTo: String): Float

**exc** 1.1 canvi-desconegut: no es disposa de conversió entre euros i cTo

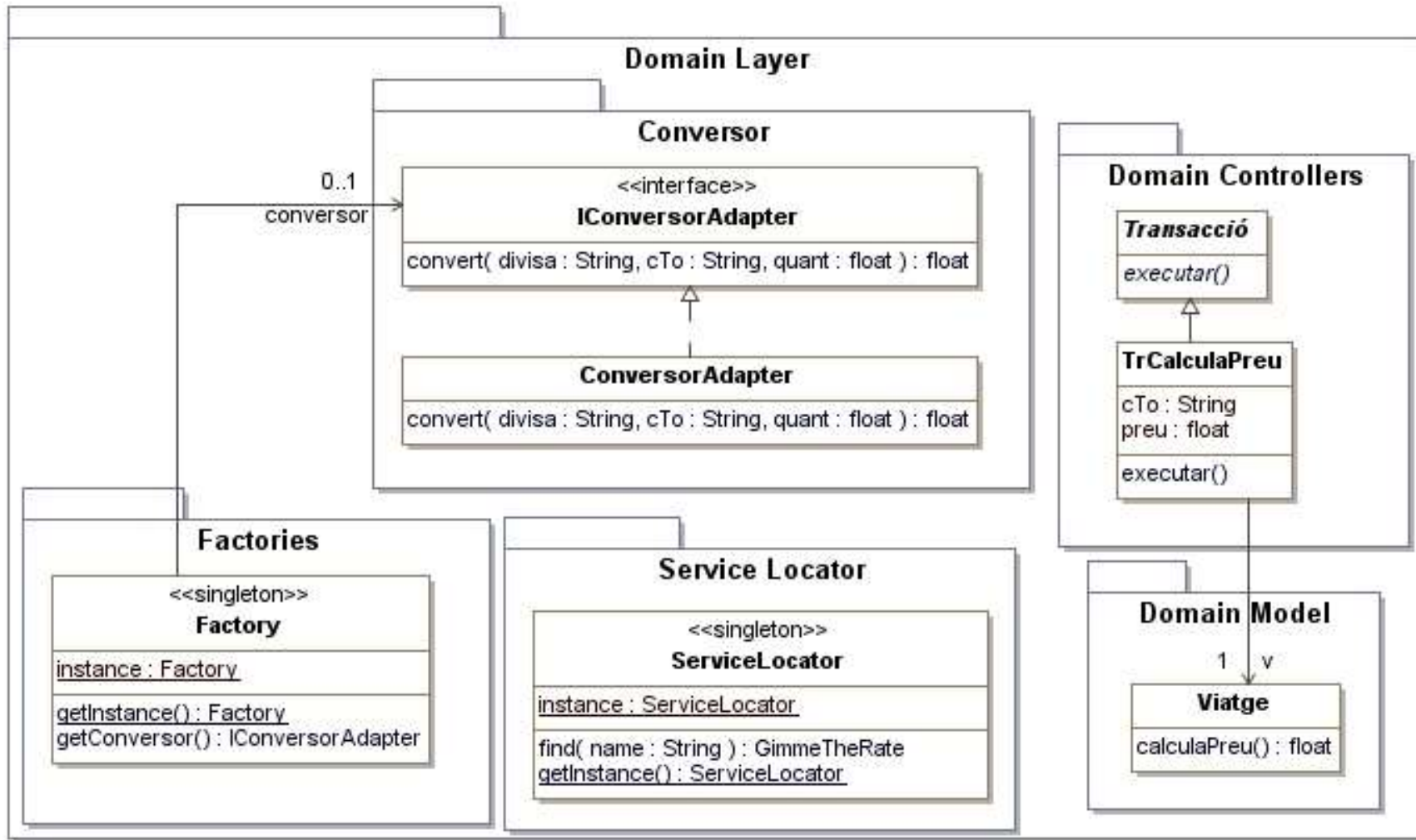
**post** 2.1: retorna el preu del viatge v expressat en la divisa cTo

**context** Viatge::calculaPreu(): Float

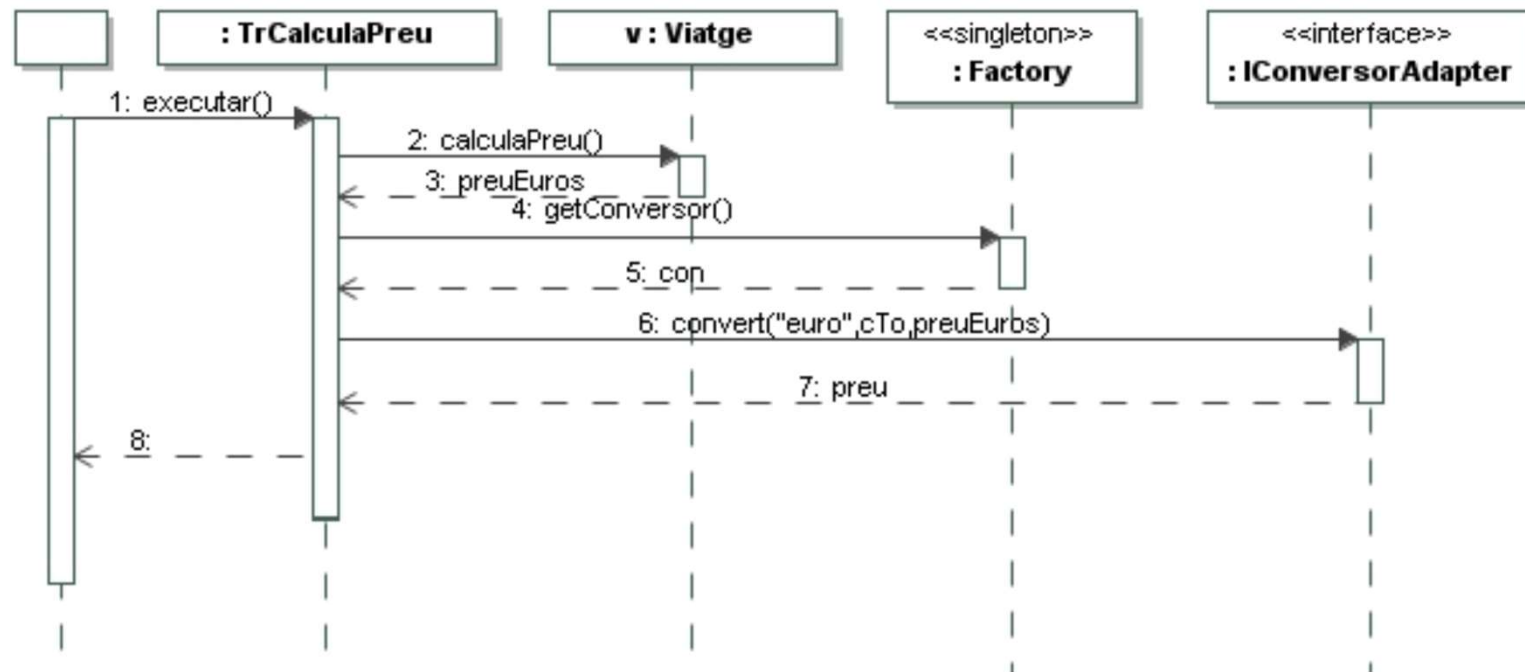
**post** 2.1: retorna el preu del viatge en euros

# Domain Layer Design – Exercise 4

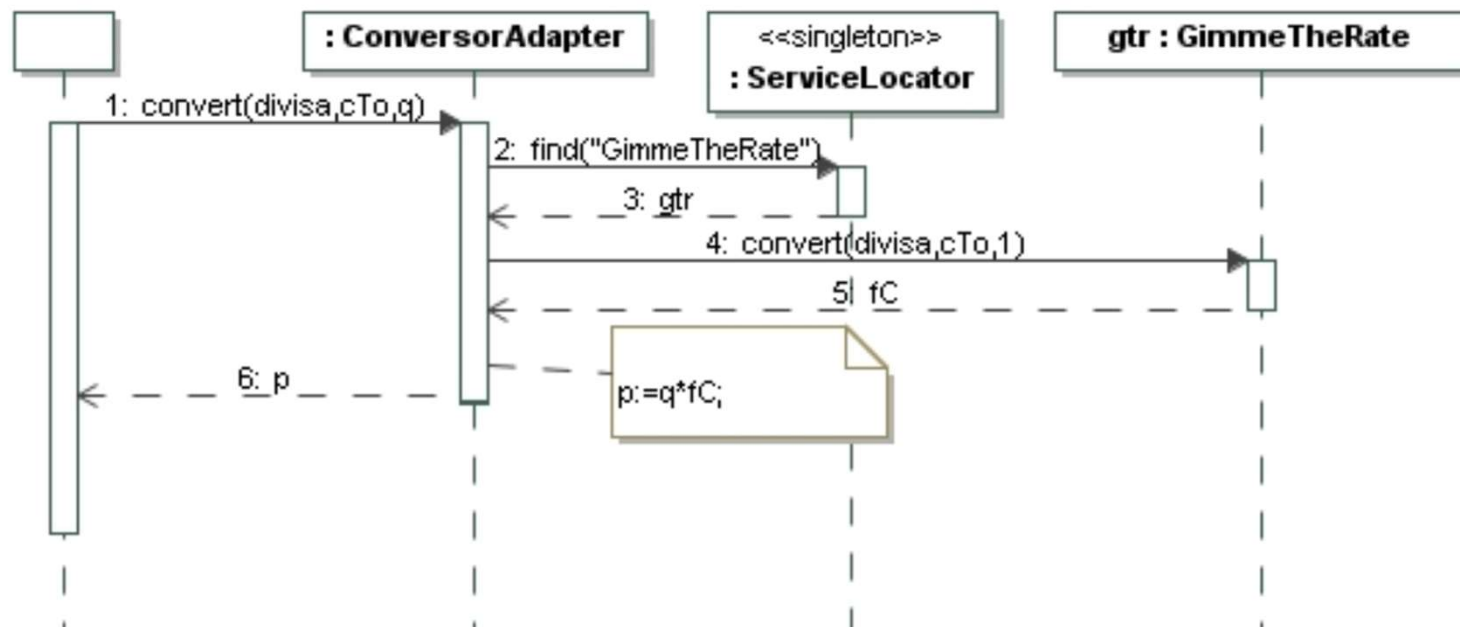
1. Dissenyeu el mètode *calculaPreu*.



## Domain Layer Design – Exercise 4

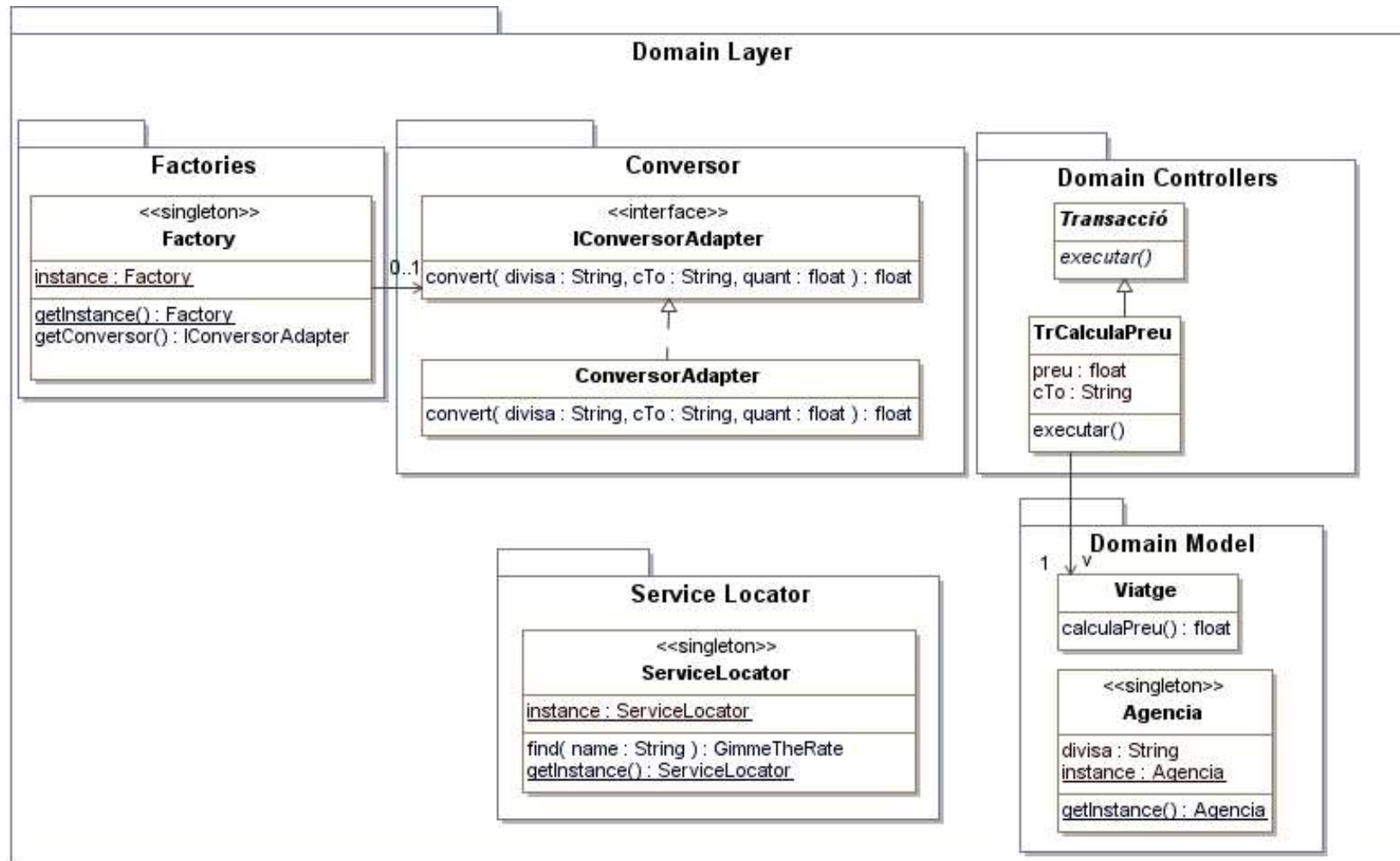


# Domain Layer Design – Exercise 4

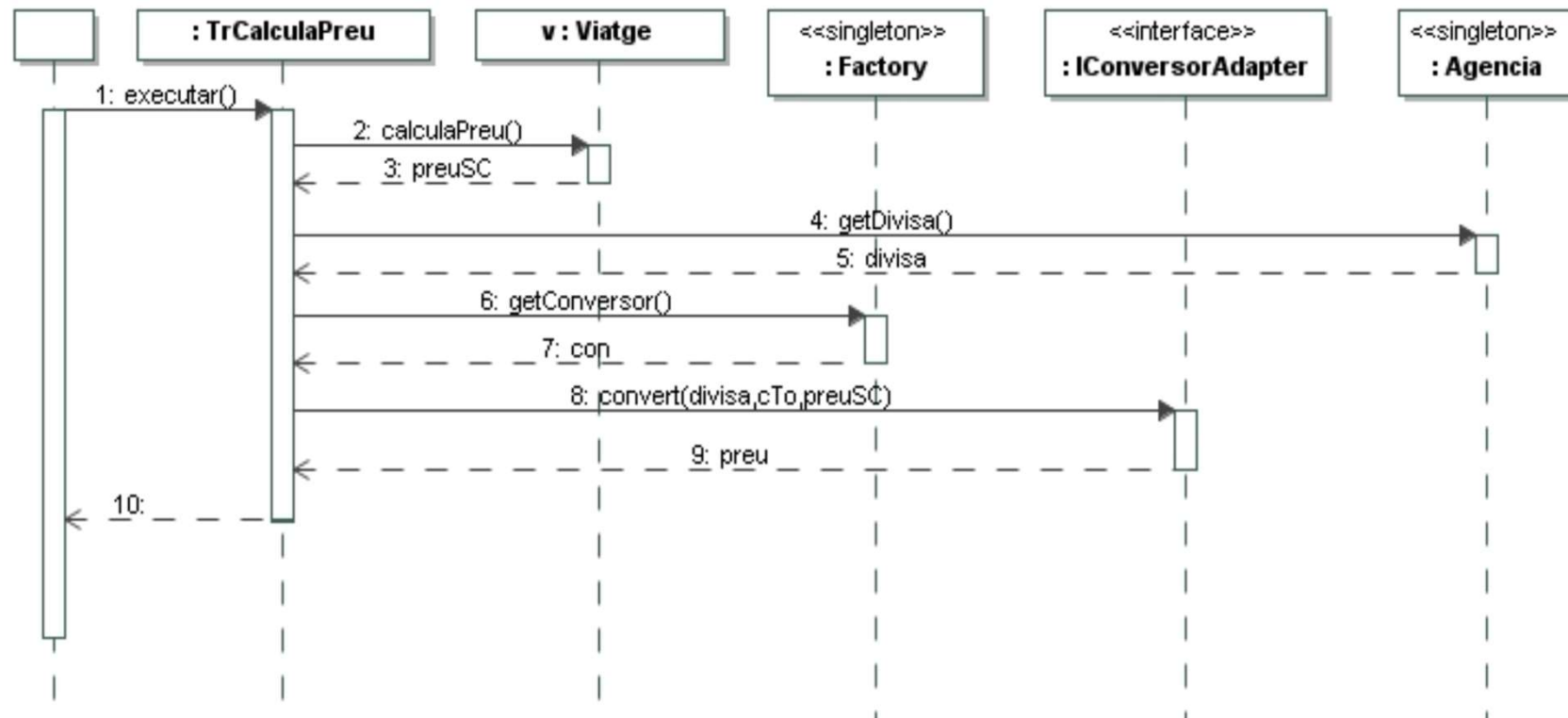


# Domain Layer Design – Exercise 4

2. Redissenyeu el mètode *calculaPreu*.



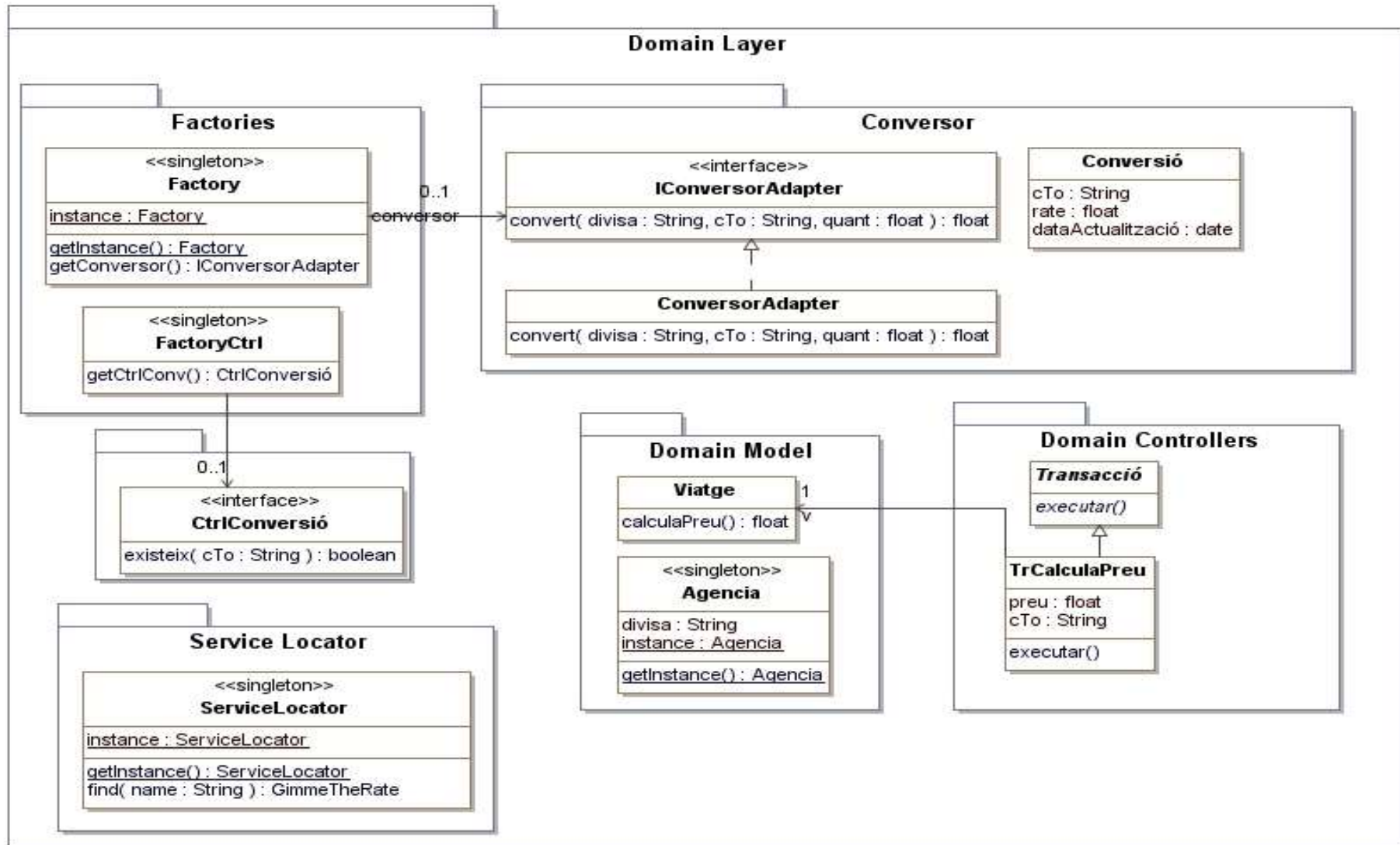
## Domain Layer Design – Exercise 4



El diagrama de seqüència de l'operació del conversor no canvia.

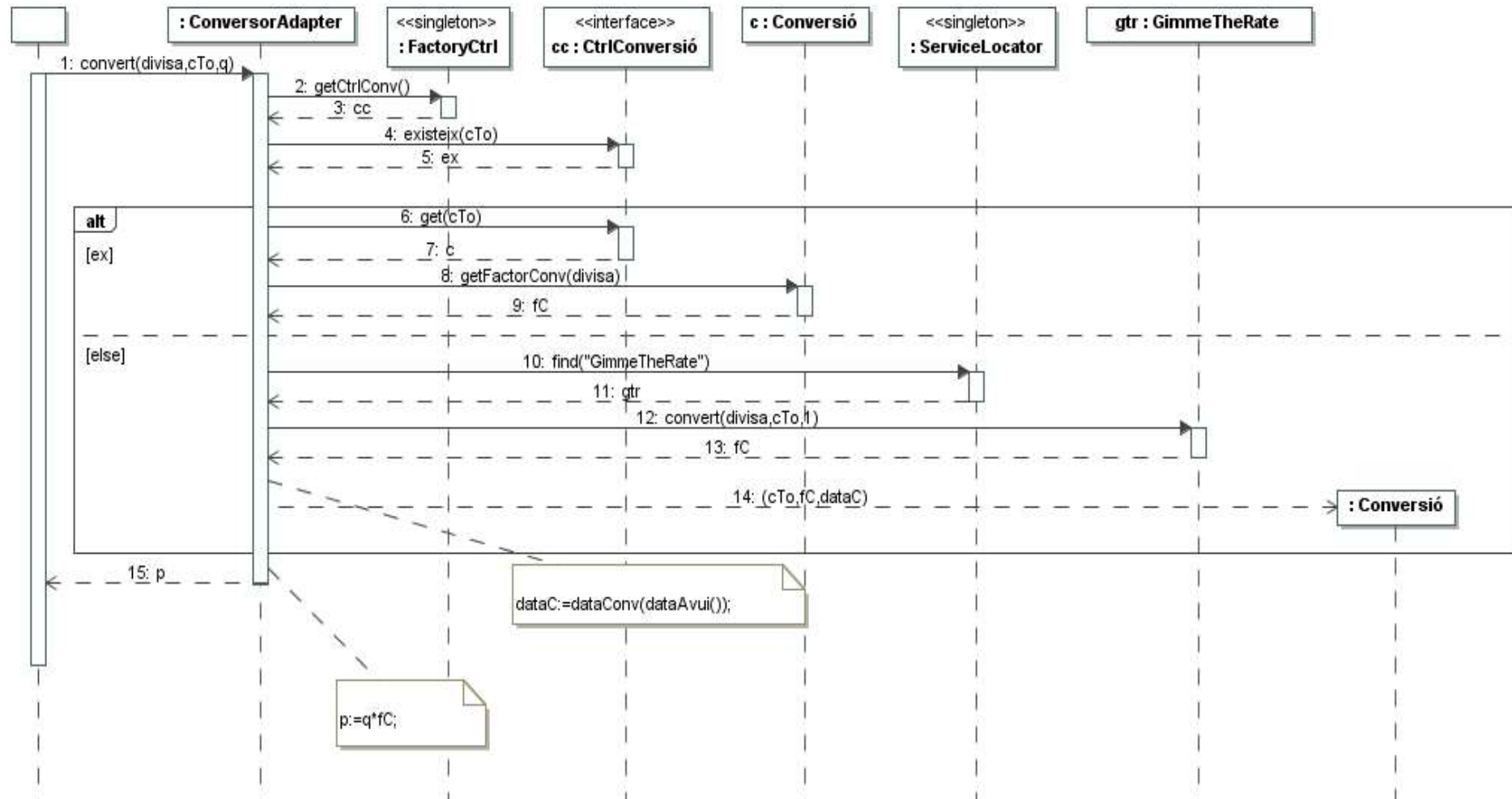
# Domain Layer Design – Exercise 4

## 3. Millora del mètode *calculaPreu*.



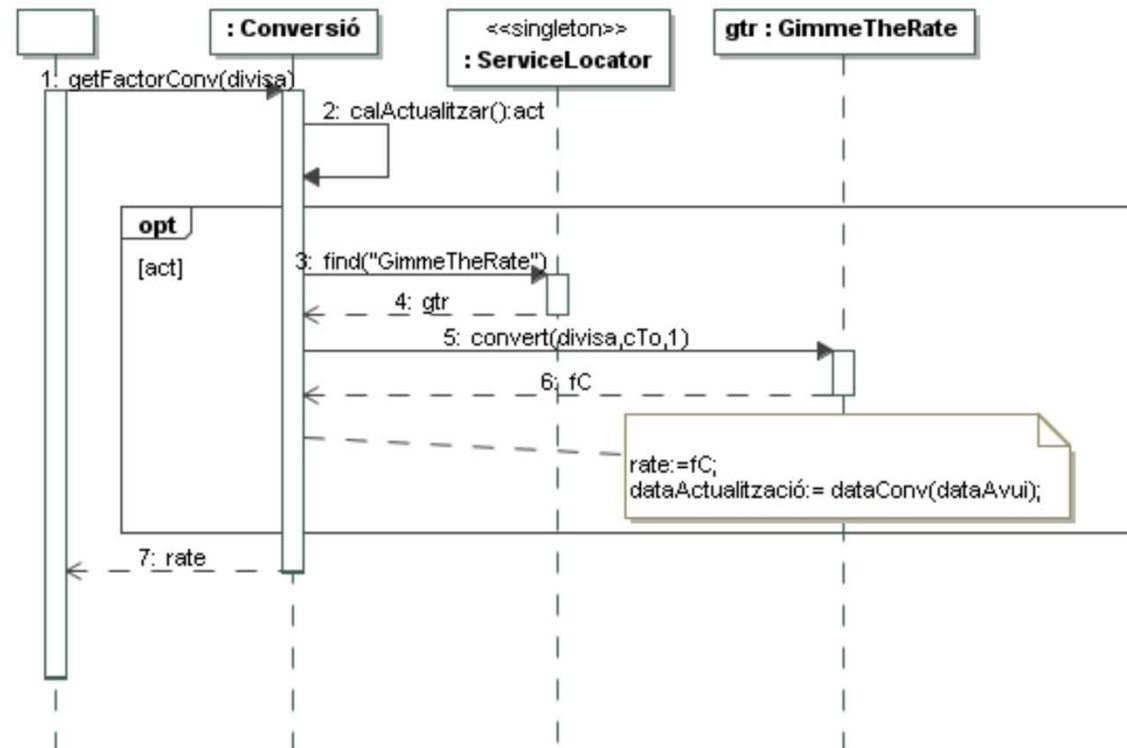


# Domain Layer Design – Exercise 4



El diagrama de seqüència de la transacció no canvia.  
*dataConv* és una operació que retorna la data (i hora) anterior a la data del paràmetre que sigui les 00:00 o 12:00

# Domain Layer Design – Exercise 4



# Domain Layer Design – Exercise 4

