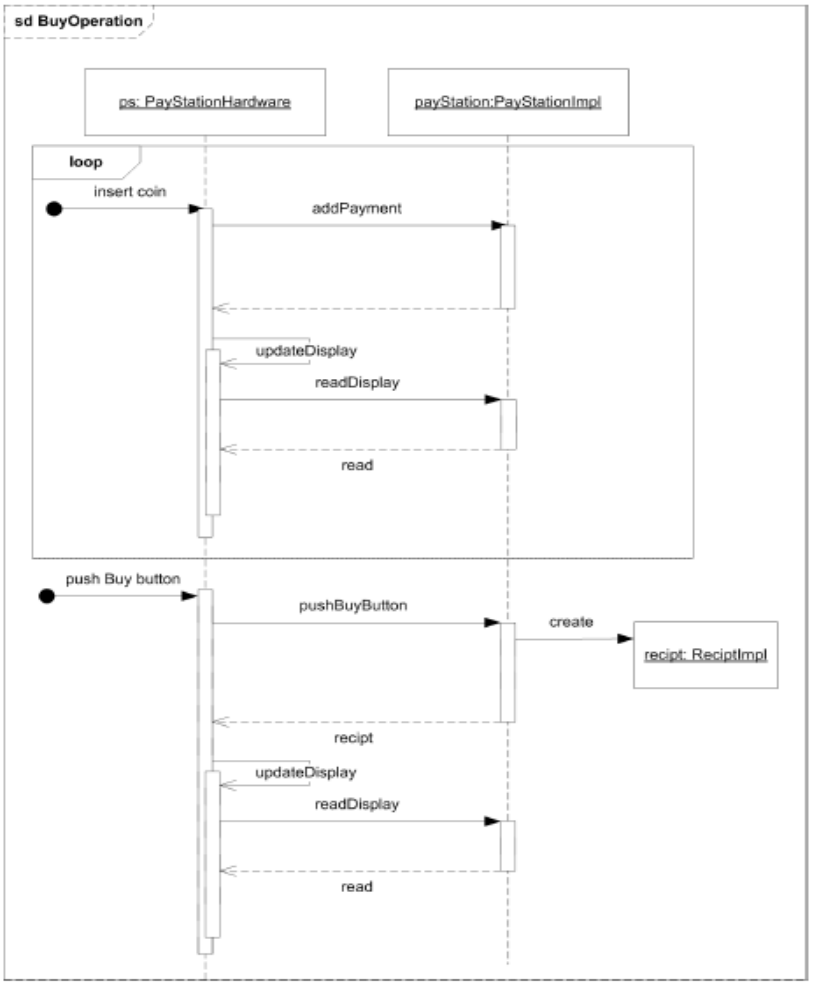
Laboratori 2

Disposem d'una versió inicial del software d'un parquímetre que està instal·lat i funcionant en dues ciutats d'Estats Units, Nova York i Boston. El parquímetre té 3 històries d'usuari implementades:

Història d'usuari 1: Comprar un tiquet de parquing. Un conductor arriba al parquímetre per comprar temps de pàrquing. El conductor introdueix diverses monedes vàlides (5, 10 i 25 centaus de dòlar). En la pantalla del parquímetre, el conductor pot veure el temps de pàrquing comprat després de la introducció de cada moneda. Quan està satisfet amb el temps comprat, el conductor prem el botó "*Buy*" i rep un rebut amb els minuts de temps comprats. La pantalla del parquímetre es torna a posar a 0 pel següent conductor. Actualment, la ciutat de Nova York aplica una tarifa lineal proporcionant 2 minuts de pàrquing per cada 5 centaus introduïts. En canvi, la ciutat de Boston aplica una tarifa progressiva d'acord amb el següent esquema (el temps proporcionat serà sempre en minuts i serà un integer):

- Primera hora: $1,5 (5 centaus proporciona 2 minuts)
- Segona hora: $2 (10 centaus proporciona 3 minuts)
- Tercera i successives hores: $3 (5 centaus proporciona 1 minut)

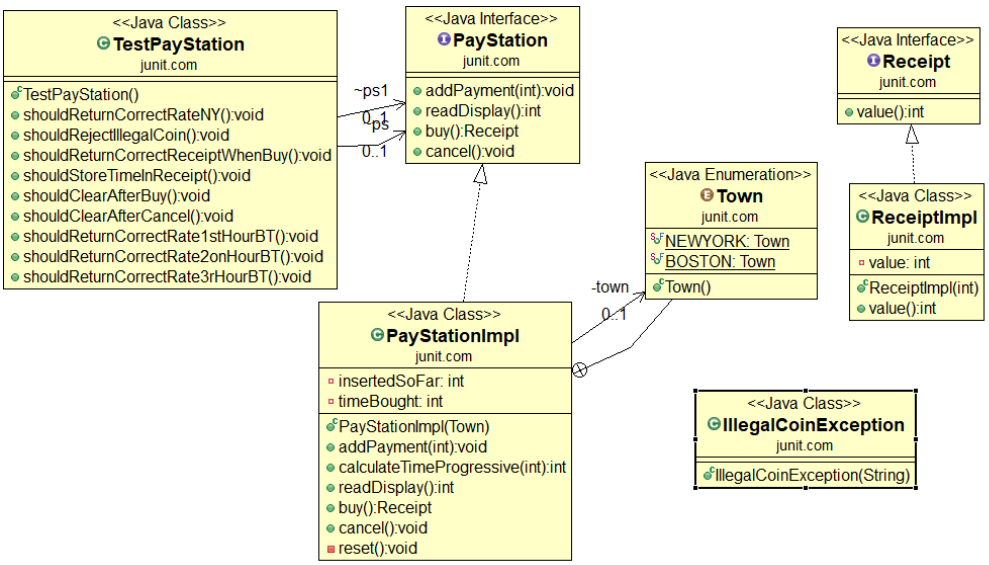
A continuació teniu un diagrama de seqüència que mostra la interacció del hardware del parquímetre amb el software que tenim desenvolupat:



Història d'usuari 2: Cancel·lar la compra d'un tiquet de pàrquing. Un conductor després d'introduir diverses monedes se n'adona que el temps de pàrquing comprat excedeix el que necessita. Per tant, prem el botó "*Cancel*" i el parquímetre retorna les monedes introduïdes. La pantalla del parquímetre es torna a posar a 0 pel següent conductor.

Història d'usuari 3: Rebutjar monedes no vàlides. Un conductor per error introdueix una moneda de 50 centaus. El parquímetre no reconeix la moneda, la rebutja i no actualitza la pantalla.

El codi i els tests corresponent a aquestes històries d'usuari els podeu trobar a l'Atenea. El diagrama de classes corresponent a aquest codi és el següent:



És possible que en el futur altres ciutats estiguin interessades en comprar el nostre software amb un càlcul de tarifes que pot ser diferent. A més, les ciutats que actualment tenen el software instal·lat poden voler canviar la tarificació, fins i tot, en temps d'execució. Ens adonem que el software actual no està preparat per aquests canvis i que, per tant, caldria refactoritzar-lo.

Es demana:

a) Penseu quin patró es podria aplicar per tal de resoldre el problema que es planteja.

b) Feu el diagrama de classes del model de domini (només la part del diagrama que canvia) amb els atributs, associacions i operacions de la solució que proposaries.

c) Implementeu el refactoring per aplicar la solució descrita.