

Nom: .....

Els ajuntaments han decidit cobrar un nou impost per gravar la contaminació dels vehicles. Ens han demanat afegir una nova història d'usuari al software PayStation (que hem estat desenvolupat a la classe de laboratori) per tal de que els usuaris dels parquímetres puguin consultar l'import d'aquest impost.

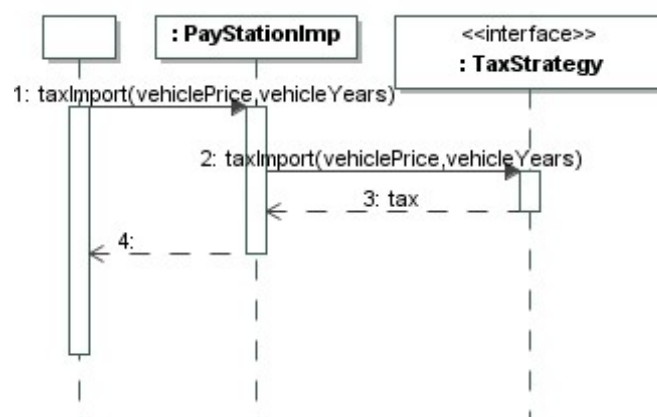
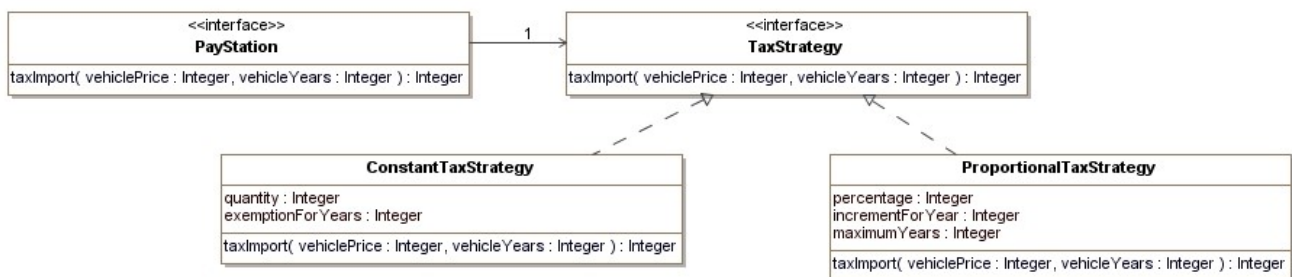
Cada ajuntament pot decidir com calcula aquest import i el càlcul pot variar en un futur ja que és un impost de nova implantació. Sabem que el càlcul de l'impost sempre dependrà de dos valors que haurà d'introduir l'usuari: el *preu* i els *anys* dels vehicle, tots dos valors enters. Inicialment, el software del parquímetre oferirà dues maneres diferents de calcular l'import de l'impost (enter):

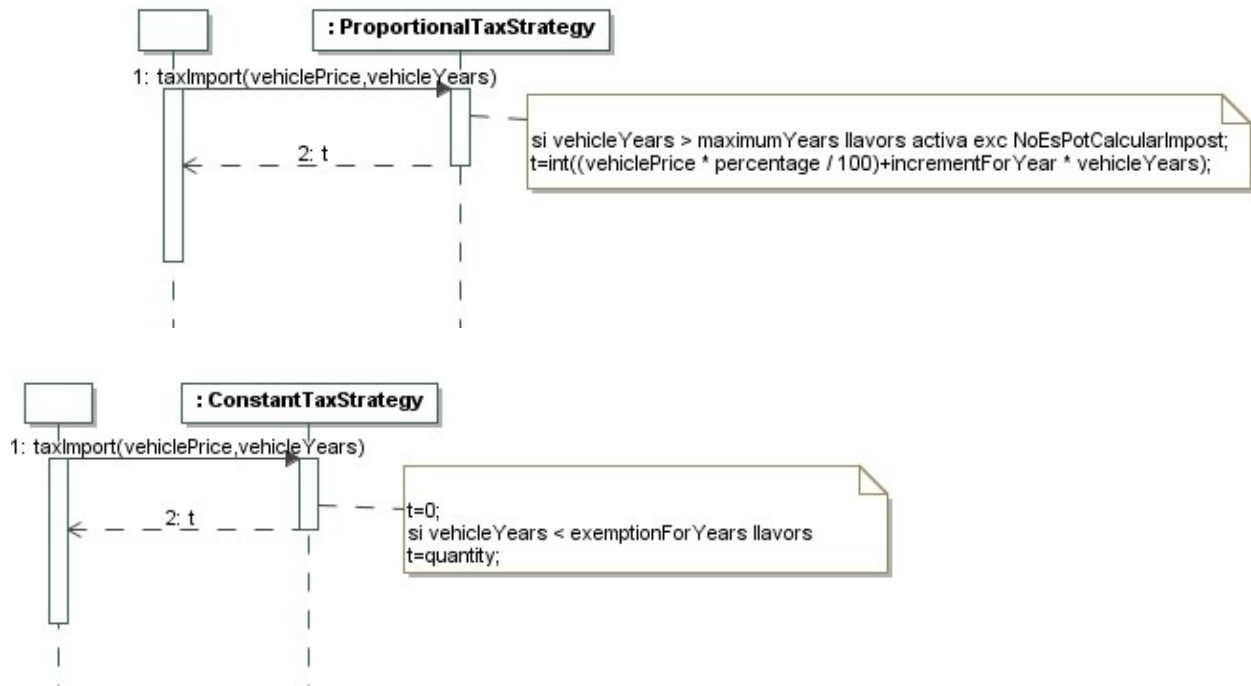
- 1) *Impost Fix*. L'import de l'impost serà una quantitat (enter) que serà fixada per l'ajuntament en donar d'alta el parquímetre. No pagaran aquest impost, és a dir, la quantitat serà 0, els vehicles que tinguin menys dels anys que estableixi l'ajuntament.
- 2) *Impost Proporcional*. L'import de l'impost dependrà de tres valors fixats per l'ajuntament en donar d'alta el parquímetre, *percentatge*, *incrementPerAny* i *anysMàxim* (enters). L'import es calcularà aplicant el *percentatge* al preu del vehicle més un *incrementPerAny* aplicat a cada any que tingui el vehicle. Si el nombre d'anys del vehicle és superior al valor *anysMàxim* fixat per l'ajuntament s'activarà l'excepció *NoEsPotCalcularImpost*.

Es demana:

**1. [3 punts] (Temps estimat: 15 minuts) (Competència transversal) Expliqueu de forma breu i raonada** la solució (patró) que proposaríeu per afegir aquesta història d'usuari al parquímetre. A més, proporcioneu la solució del punt de vista estàtic (**diagrama de classes amb els atributs i operacions**) com des del punt de vista del comportament (**diagrames de seqüència**). Volem una solució que: 1) permeti als parquímetres seleccionar una forma de càlcul de l'impost a l'inici i canviar-la en temps d'execució i 2) permeti afegir noves formes de càlcul (sense fer canvis, només extensions). No cal fer servir el patró factoria. **Contesteu aquesta pregunta a l'espai que teniu disponible.**

Utilitzariem una solució on aplicariem el patró estratègia. El comportament variable del càlcul de l'impost quedaria encapsulada en una interface i les classes que implementarien aquesta interface implementarien el càlcul concret que han de tenir els parquímetres. A continuació es mostra el diagrama de classes per aquesta solució:





**2. [3 punts] (Temps estimat: 20 minuts)** Expliqueu clarament les iteracions que seguireu per implementar aquesta nova història d'usuari al software PayStation utilitzant TDD. Cal que quedi clar el que fareu a cada iteració. **Contesteu aquesta pregunta a l'espai que teniu disponible.**  
(Modificar solució)

- Iteració 1:
  - Definir test per calcular l'import de l'impost que retorni un fake.
  - Definir el codi per tal de que passi el test.
- Iteració 2 (Refactoring):
  - Introduir la interface TaxStrategy
  - Refactoritzar PayStationImpl per introduir la referència a la interface TaxStrategy i usar la referència per fer el càlcul (comprovar que el test corresponent falla ja que la interface no té cap classe que la implementa).
  - Definim una estratègia SimpleTaxStrategy simple (retorna un fake) que implementi la interface TaxStrategy. La implementació de l'estratègia s'ha de definir en la carpeta de test ja que el codi d'aquesta estratègia no estarà dins del codi de producció.
  - Refactoritzar l'operació de setUp per tal de que en crear l'objecte PayStationImpl usi l'estratègia simple.
- Iteració 3 (Estratègia constant): Definim un nou test case, TestConstantStrategy per definir els tests relacionats amb aquest càlcul.
  - Definim el test pel cas de l'import 0.
  - Definim el codi per tal de que el test anterior passi.
- Iteració 4 (Estratègia constant):
  - Definim el test pel cas de l'import més gran que 0.
  - Definim el codi per tal de que el test anterior passi.
- Iteració 5 (Estratègia proporcional): Definim un nou test case, TestProportionalStrategy per definir els tests relacionats amb aquest càlcul.
  - Definim el test pel cas anys del vehicle inferior al valor *anysMàxim*.
  - Definim el codi per tal de que el test anterior passi.
- Iteració 6 (Estratègia proporcional superior anysMàxim):
  - Definim el test pel cas anys del vehicle superior al valor *anysMàxim*.
  - Definim el codi per tal de que el test anterior passi.
- Iteració 7 (Test Integració):

- Definir un test case TestIntegration per provar que els pay stations amb les dues formes de càlcul funcionen tal i com s'espera.