

The background of the entire page is a green circuit board pattern. It features a complex network of white lines representing circuit traces, with various electronic components like chips and pads depicted in white. The pattern is dense and covers the entire area.

DAM 1

Exercici 52_16.

Documentem l'extracció

Soufian El Hajouji

ÍNDEX

Pregunta 1. El meu exemple Refactoring.....	2
---	---

Exercici 52_12. Documentació de la refacció

Enunciat

En aquest exercici començarem a documentar els canvis que realitzarem al nostre projecte.

Crea un document una mica formal (amb portada, taula de continguts, secció d'introducció i conclusions) i afegeix una secció on incloguis el diagrama de classes, i el de seqüència d'inici del projecte de refacció.

Pots aprofitar la secció de conclusions per descriure les teves reflexions respecte l'estat actual del projecte.

Aquest document l'aniràs completant a mida que vagis avançant en els exercicis.

Introducció

Aquest document descriu el procés de refacció d'un sistema de gestió de lloguer de vehicles. Aquest sistema ja existeix i ha estat desenvolupat però, degut a les necessitats dels usuaris, es requereix una revisió completa i una millora en la seva estructura per aconseguir una millor eficiència i mantenibilitat.

El sistema actual consisteix en tres classes: Client, Lloguer i Vehicle. La classe Client conté informació dels clients com el NIF, el nom i el telèfon. La classe Vehicle conté informació dels vehicles com la marca, el model i la categoria, i la classe Lloguer conté informació sobre els lloguers, incloent el vehicle llogat i la durada del lloguer.

Diagrama de Classes

El diagrama de classes del sistema de gestió de lloguer de vehicles és el següent:

```
class Vehicle {  
    + {static} BASIC = 1: int  
    + {static} GENERAL = 2: int  
    + {static} LUXE = 3: int  
    - model: String  
    - marca: String  
    - categoria: int  
    + Vehicle(marca, model, categoria)  
}
```

```
class Lloguer {  
    - dies: int  
    + Lloguer(vehicle, dies)  
}
```

```
class Client {  
    - nif: String  
    - nom: String  
    - telèfon: String
```

```

- lloguers: ArrayList<Lloguer>
+ Client(nif, nom, telefon)
+ getNif(): String
+ getNom(): String
+ getTelefon(): String
+ informe(): String
}

```

```

Vehicle <-left- Lloguer
Lloguer <-left- Client: *

```

Diagrama de Seqüència

El diagrama de seqüència a continuació mostra la seqüència d'inici del projecte de refacció, que comença amb un usuari que sol·licita un informe sobre un client.

```

actor Usuari
Usuari -> unClient: informe()
unClient -> unLloguer: getVehicle()
note left: per cada lloguer
unClient -> unVehicle: getCategory()
unClient -> unLloguer: getDies()

```

Conclusions

En resum, el sistema de gestió de lloguer de vehicles existent necessita una refacció completa per millorar la seva eficiència i mantenibilitat. Els diagrames de classes i seqüència són una eina útil per ajudar en la planificació del projecte de refacció, i el diagrama de classes mostra les classes que s'han de desenvolupar i les seves relacions.

Exercici 52_16. Documentem l'extracció

Enunciat

Afegeix una nova secció a la documentació on descriuràs els canvis realitzats al codi amb el moviment de mètodes de l'exercici anterior.

Aquest canvi segurament ha modificat el diagrama de classes i/o de seqüència. Redibuixa els diagrames que hagin canviat perquè reflexin la nova situació.

Aprofita la secció de conclusions per descriure les teves reflexions respecte aquest canvi.

s'ha afegit un nou mètode a la classe Client anomenat "quantitatPerLloguer", que permet saber el preu que ha de pagar cada client pel lloguer. A més, s'ha omplert el mètode

"informe" a la mateixa classe, que proporciona informació detallada sobre els lloguers realitzats per un client determinat.

El diagrama de classes també s'ha actualitzat per reflectir aquestes noves funcionalitats. Ara, la classe Client té un mètode addicional i la classe Lloguer té una relació bidireccional amb la classe Client.

Finalment, el diagrama de seqüència també s'ha actualitzat per reflectir la crida al mètode "quantitatPerLloguer" per part del client. Això és important perquè ara el client pot saber el preu que ha de pagar per cada lloguer realitzat.

Diagrama de Classes actualitzat:

```
class Vehicle {
+ {static} BASIC = 1: int
+ {static} GENERAL = 2: int
+ {static} LUXE = 3: int
- model: String
- marca: String
- categoria: int
+ Vehicle(marca, model, categoria)
}

class Lloguer {
- dies: int
+ Lloguer(vehicle, dies)
}

class Client {
- nif: String
- nom: String
- telèfon: String
- lloguers: ArrayList<Lloguer>
+ Client(nif, nom, telefon)
- quantitatPerLloguer(Lloguer): double
+ informe(): String
}

Vehicle <-left- Lloguer
Lloguer <-left- Client: *

Vehicle <-left- Lloguer
Lloguer <-left- Client
```

Diagrama de Seqüència actualitzat:

```
actor Usuari
Usuari -> unClient: informe()
unClient -> unLloguer: getVehicle()
note left: per cada lloguer
unClient -> unVehicle: getCategoria()
unClient -> unLloguer: getDies()
unClient -> unClient : quantitatPerLloguer()
```