



JESUÏTES El Clot
Escola del Clot

M03. Programació II

UF6. POO. Introducció a la persistència en BD Projecte

Curs 2021-2022



CLASSICS CARS RENTAL



Contenido

EXERCICI 1. Aplicació en Java per gestionar la empresa CLASSICS CARS RENTAL	3
Introducció.....	3
Operacions que ha de permetre fer l'aplicació (CRUD).....	4
Fases del projecte	4
Fase 1: Requeriments funcionals.....	4
Fase 2: Anàlisi i disseny	4
Fase 3: Implementació de l'aplicació	4
Tenir en compte.....	5
EXERCICI 2. El patró DAO	5
Introducció.....	5
Pregunta 1.....	6
Pregunta 2.....	6
Pregunta 3.....	6
LLIURAMENT	6
CRITERIS D'AVUACIÓ.....	7
RUBRICA.....	7

EXERCICI 1. Aplicació en Java per gestionar la empresa CLASSICS CARS RENTAL

Introducció

Una petita empresa de lloguer de cotxes antics, de nom **CLASSICS CARS RENTAL**, ens han demanat que els desenvolupem un prototipus d'aplicació d'escriptori basada en Java que els permeti millorar la gestió de la seva empresa. Així doncs, per parelles, haureu d'especificar, dissenyar i implementar un prototipus de l'aplicació, al més aviat possible, què permetre dur a termini la gestió del lloguer dels cotxes. Convindrà determinar quina informació serà necessària. Per exemple:

- Dels **cotxes** que té la empresa, s'haurà que emmagatzemar la matrícula, número de bastidor, marca, model, color, nombre de places, nombre de portes, grandària del maleter, tipus de combustible (gasolina/diesel/elèctric), etc. etc
- Dels **clients** que llogant els cotxes, s'haurà que emmagatzemar dades com el Dni, nom i cognoms, edat, telèfon, adreça, ciutat, país, email, número del permís de conducció, punts del permís de conducció, etc. etc.
- Del **lloguer dels cotxes** s'haurà que emmagatzemar dades com DNI del client que lloga el cotxe, matrícula del cotxe que ha llogat data del préstec, per quants dies, preu per dia, lloc per fer la devolució del cotxe, préstec amb retorn de dipòsit ple (sí/no), Tipus d'assegurança (amb franquícia o sense franquícia)
- Dels **mecànics** que s'encarreguen del manteniment dels cotxes, s'haurà de emmagatzemar dades com el DNI, nom i cognoms, edat, telèfon, adreça, ciutat, país, email, número del permís de conducció, punts del permís de conducció, salari, nombre de la seguretat social, titulació, anys que porta a la empresa.
- Del **manteniment dels cotxes** s'haurà emmagatzemar dades com el dni del mecànic que fa el manteniment del cotxe, la matricula del cotxe, data de inici i data de fi del manteniment del cotxe, etc. etc
- Tenir en compte que, en aquesta empresa, es permet que un client llogui més d'un cotxe però un cotxe només pot ser llogat per un client. Un mecànic pot fer el manteniment de diversos cotxes però un cotxe només pot ser revisat per un mecànic.
- Els mecànics No son clients de la empresa

Operacions que ha de permetre fer l'aplicació (CRUD)

1. Alta / Modificació / Eliminació / Llistat / Cerca de CLIENTS (per dni)
2. Alta / Modificació / Eliminació / Llistat / Cerca de COTXES (per matricula o per tipus de cotxe)
3. Alta / Modificació / Eliminació / Llistat / Cerca (per dni) de MECÀNICS
4. Alta / Modificació / Llistat / Cerca/Gestió del LLOGUER DE COTXES
5. Alta / Modificació / Llistat / Cerca/Gestió del MANTENIMENT DELS COTXES
6. Mostrar els cotxes que estan llogats, així com el dni, nom i telèfon de la persona que ho ha llogat i per quants dies

Fases del projecte

Fase 1: Requeriments funcionals.

Redactar un document amb el que farà exactament la teva aplicació, assignant un número de requeriment a cadascun dels requeriments funcionals que us surtin, de la manera més ordenada, clara i menys ambigua possible. Numerar els requeriments amb un codi mnemotècnic específic, per exemple:

RQ001 -

RQ002 -

Fase 2: Anàlisi i disseny

- El modelo relacional associat a la base de dades que vas a crear
- Creació de la base de dades en MySql

Fase 3: Implementació de l'aplicació

Utilitzar el paradigma de la POO amb Java

- La interacció amb l'aplicació és en mode consola per tant, crear la classe principal amb un menú què permeti gestionar les operacions que es demanen. El que vulgui per utilitzar JavaFx

- Programació de la persistència de l'aplicació. Utilitzarem JDBC per a realitzar totes les operacions necessàries envers la BD (connexió amb la base de dades i el CRUD).
- Fer una classe **Connectar** on es posi la connexió a la BD
- Utilitzar sentències preparades (**Prepared Statement**) amb tots els casos que siguin necessaris
- Crear un mètode per a cada acció del menú i guardar-los en una classe de nom **ClasseCRUD**
- Demanar dades per teclat quan sigui necessari
- Utilitzar correctament la gestió d'errors (tractament amb excepcions predefinides i pròpies), Per exemple controlar que en el menú no es posi un número que no correspongui; no deixar entrar números ni caràcters especials en el cas que sigui una cadena de caràcters com a nom, cognoms, ... i no deixar entrar lletres o caràcters especials en el cas que sigui un número
- Controlar tots els requeriments prèviament definits al document de RQ.

Tenir en compte

Els següents aspectes són de caràcter **OBLIGATORI, EN CAS CONTRARI NO ES CONSIDERARÀ VÀLID EL PROJECTE, TOT I QUÈ FUNCIONI:**

- El lliurament del model i la creació de la base de dades
- La connexió de Java amb MySql
- Llistar la flota de cotxes de la empresa
- Poder llogar un cotxe

EXERCICI 2. El patró DAO

Introducció

Existeixen molts patrons de disseny, per exemple els patrons MVC, DAO, DTO són els més utilitzats en Java. Cadascun s'especialitza a resoldre un problema específic
El patró DAO (Data Acces Object) resol el tema d'accés a les dades. Suposem que tens un

sistema muntat amb una base de dades MySQL i l'hagis de canviar a PostgreSQL o qualsevol altre motor de base de dades; això pot ser un veritable problema. Precisament això és el que ajuda a solucionar aquest patró DAO, tenir una aplicació que no estigui lligada a l'accés a dades, que si, per exemple, la part de la vista demana trobar els clients amb salaris mensuals majors 1000, el DAO s'encarregui de portar aquestes dades independentment si està en MySQL o en PostgreSQL o qualsevol altre SGBD. La capa DAO conté tots els mètodes CRUD (create, read, update, delete). De manera general es té un DAO per a cada taula en la base de dades

Pregunta 1.

Explicar les avantatges d'un patró de disseny

Pregunta 2.

Explicar, amb més detalls, en què consisteix el patró DAO

Pregunta 3.

Fer un exemple **senzill** emprant el patró DAO

LLIURAMENT

- **Presentació del projecte: data màxima 22 d'abril de 2022, dins de l'horari de classe.**
- Cal lliurar el projecte abans de la data d'entrega que trobareu al moodle
- S'ha d'entregar en fitxer zip o tar, amb el següent format:
 - *COGNOM1_COGNOM2_PROJECTE_UF6.zip*
 - Aquest fitxer comprimit ha de contenir el codi de la creació de la base de dades, el codi font de l'aplicació i codi fon del exemple amb el patró DAO.
 - Un pdf amb els requeriment, els diagrames i les captures de pantalla de les funcionalitats de l'aplicació. Ha de tenir portada i estar ben editat.
 - El lliurament només l'ha de fer un dels components del grup.

CRITERIS D'AVUACIÓ

- La puntuació màxima és un 10.
- La puntuació mínima és de 5 sobre 10.
- Hi ha 0,5 punts extra si s'utilitza JavaFx
- Cal lliurar el projecte abans de la data especificada al moodle

RUBRICA