



**UT5**  
José R. Mas Davó authored 3 weeks ago

85519756

 **ProjecteITV-POO.md** 7.67 KiB

# Taller d'ITV - 1. Introducció a la Programació Orientada a Objectes

## 1. Introducció

### 1.1. Context i justificació

En aquesta primera fase del projecte, es pretén implementar les bases de l'aplicació "Taller d'ITV", utilitzant els conceptes fonamentals de la programació orientada a objectes (POO). Es treballarà amb classes, objectes, atributs i mètodes per establir una estructura sòlida sobre la qual es desenvoluparan les funcionalitats avançades en fases posteriors.

### 1.2. Objectius generals d'aquesta fase

Implementar les classes bàsiques per a la gestió de vehicles i clients, així com les funcions principals de recepció i gestió de cues en un taller d'ITV.

### 1.3. Competències a desenvolupar

- Disseny i implementació de classes en Java.
- Creació i manipulació d'objectes.
- Encapsulament i interacció entre objectes.
- Implementació de mètodes i atributs.
- Gestió bàsica d'entrada/eixida per consola.

### 1.4. Pregunta guia

Com es pot dissenyar una estructura de classes per a la gestió d'un taller d'ITV utilitzant principis de programació orientada a objectes?

## 2. Descripció del repte

### 2.1. Situació inicial

Un taller d'ITV necessita gestionar els vehicles que arriben per a la revisió, incloent-hi la recepció de dades dels vehicles, la gestió de les cues i la distribució als boxs per a les revisions.

### 2.2. Necessitats a cobrir

- Registre i gestió de vehicles.
- Control de cues i boxs.
- Gestió de les fases de revisió per a cada vehicle.

### 2.3. Restriccions i consideracions

El taller de revisió es compon de 6 boxs, tots ells amb les mateixes responsabilitats de revisió de vehicles, és a dir, indistintament del box pel qual un vehicle passe, passarà per les mateixes fases de revisió, que són sempre 4 i amb el següent ordre:

1. Revisió inicial d'elements de seguretat
2. Revisió del sistema elèctric
3. Revisió d'emissió de fums
4. Revisió de frens i direcció.

Des del sistema es tindrà la capacitat decidir en qualsevol moment que els vehicles que estiguen dins un box es moguen tots alhora, de la fase de revisió on estiguen cap a la fase següent. S'assumeix en este cas que estan tots llestos per a passar de fase i que un suposat mecànic els ha donat el vistiplau.

Quan un vehicle arriba al taller se li pren tant el número de matrícula (4 números seguits de 3 lletres) com el model i del vehicle, a més del tipus que pot ser (cotxe, microbús, furgoneta o camió). Si el registre del vehicle és correcte (no hi pot haver dos vehicles amb la mateixa matrícula), este vehicle es posarà automàticament a la cua general (cua inicial).

La cua general no pertany a cap box concret, podent passar a la cua de fases de revisió del box que des del programa informàtic es

decidisca. Lògicament, la primera fase de revisió del box que reclame al següent vehicle de la cua general ha d'estar lliure per atendre este vehicle, sinó se li denegarà eixe reclam.

### 3. Planificació i organització

#### 3.1. Formació d'equips

Treball en grups per a dissenyar i implementar les diferents classes i mètodes necessaris.

#### 3.2. Cronograma i fites

Heu de proposar un cronograma orientatiu de les tasques que heu pensat desenvolupar fins a la data de lliurament. Heu de cobrir el disseny de classes, la implementació de la funcionalitat i les proves.

#### 3.3. Recursos necessaris

- IDE Java (preferiblement NetBeans)
- Documentació de Java sobre POO
- Enunciat del projecte i materials de suport proporcionats pel professor

### 4. Fase d'investigació

#### 4.1. Identificació de coneixements previs

Revisió dels conceptes bàsics de POO, com classes, objectes, atributs encapsulament i mètodes.

#### 4.2. Fonts d'informació a consultar

- Apunts de classe sobre POO.
- Documentació oficial de Java.
- Tutorials en línia sobre implementació de sistemes de cues en Java amb arrays.

### 5. Desenvolupament del projecte

#### 5.1. Tasques específiques a realitzar

- Dissenyar i implementar les classes principals: Taller, Vehicle, Box...
- Implementar la lògica per a la gestió de cues i fases de revisió.
- Desenvolupar un menú d'opcions amb les funcionalitats requerides.
- Implementar la lògica per a cada opció del menú.
- Realitzar proves i depuració del codi.

#### 5.2. Aplicació de coneixements i habilitats

Utilitzar els conceptes de POO per estructurar el codi de manera eficient i modular.

#### 5.3. Creació del producte o solució

Les opcions que el programa ha de presentar al menú com a mínim són les següents:

1. **Alta i recepció de vehicles:** s'encarregarà de demanar les dades de registre del vehicle i posar-lo automàticament a la cua general. No es podran introduir a la cua dos vehicles amb la mateixa matrícula. Si alguna de les dades s'introdueix incorrectament, es tornarà a demanar la dada fins que s'introduísca bé. Cal informar l'usuari sobre en quina posició de la cua general es troba el vehicle rebut.
2. **Reclamar vehicle per a entrar al box:** es preguntarà pel box que desitja rebre el següent vehicle que es troba a la cua inicial, si és que n'hi ha algun i hi ha lloc per a ell en eixe box (no hi ha d'haver cap vehicle en la primera fase de revisió d'eixe box). Si no s'introdueix un número de box correcte, es tornarà a demanar fins que siga vàlid. Si no és possible l'entrada del vehicle al box, es denegarà esta operació informant a l'usuari (tant en el cas favorable com al desfavorable de l'operació).
3. **Moure tots els vehicles de fase dins d'un box.** Es preguntarà pel número de box on es vol desplaçar una fase a tots els vehicles dins seu. Per descomptat, si un vehicle es troba a l'última fase, quedarà ja fora del box.
4. **Informació de l'estat d'un box concret:** es preguntarà pel número de box (entre 1 i 6). Es mostrarà en quin estat es troba cada fase del box, mostrant de cadascuna quin vehicle té assignat (mostrant la seua matrícula, marca, model i el seu tipus), si el té.
5. **Informació general de tots els boxes:** similar a l'opció anterior però pels 6 boxes del taller.
6. Eixir del programa.

#### 5.4. Revisió i millora contínua

Proves de funcionalitat i revisió del codi per identificar i corregir errors.

### 6. Criteris d'avaluació

#### 6.1. Rúbrica d'avaluació

Es valorarà:

- Correcta implementació de les classes i objectes.
- Funcionalitat del sistema de cues i gestió de vehicles.
- Qualitat i claredat del codi.
- Compliment de tots els requisits especificats.

## 6.2. Entregables esperats

- Codi font complet del projecte NetBeans amb documentació (javadoc) de classes i mètodes
- Diagrama de classes.
- Breu informe explicant l'estructura del projecte i les decisions de disseny.

## 6.3. Ponderació de cada part

Valors assignats per a cada aspecte del projecte:

1. Alta i recepció de vehicles: 15%
2. Reclamar vehicle per a entrar al box: 20%
3. Moure tots els vehicles de fase dins d'un box: 25%
4. Informació de l'estat d'un box concret: 5%
5. Informació general de tots els boxes: 5%
6. Disseny de classes: 10%
7. Qualitat del codi i documentació: 10%
8. Presentació: 10%

# 7. Presentació

Els alumnes hauran de fer una breu demostració del funcionament de la seua aplicació, explicant les parts més rellevants.

# 8. Reflexió i metacognició

## 8.1. Preguntes per a l'autoavaluació

Què hem après sobre la programació orientada a objectes? Quins reptes hem superat?

## 8.2. Anàlisi del procés d'aprenentatge

Reflexió sobre les habilitats desenvolupades i les dificultats trobades.

## 8.3. Proposta de millores per a futurs projectes

Recomanacions per a futures fases del projecte, basades en l'experiència obtinguda.