

Découverte

Gestion des objets POO : Trains

Rédigé par

David ROUMANET
Professeur BTS SIO



Changement

| Date | Révision |
|------|----------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Sommaire

| | |
|------------------------------|-----|
| A Introduction..... | . 1 |
| A.1 Présentation..... | . 1 |
| A.2 Prérequis..... | . 1 |
| B Titre N°2..... | . 2 |
| C Annexes..... | . 3 |
| C.1 Sources..... | . 3 |
| C.2 Autres..... | . 3 |
| D Ce qu'il faut retenir..... | . 5 |

Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.
- **Découvrir** : Travaux dirigés. Faisable sans matériel.
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

A Introduction

La compréhension des notions de programmation orientée objet implique de pratiquer et de se heurter à la réalité de fonctionnement des objets.

A.1 Présentation

Nous allons programmer la gestion d'un système ferroviaire, avec des trains, des conducteurs et des gestionnaires.

L'application sera très simple, pour que la notion pratique soit la compréhension des objets et pas autre chose.

A.2 Cahier des charges

Le principe est de créer des trains, de les conduire d'un point A (source) à un point B (destination) par un conducteur (nom et matricule).

Seuls les matricules supérieurs à 1000 sont des gestionnaires.

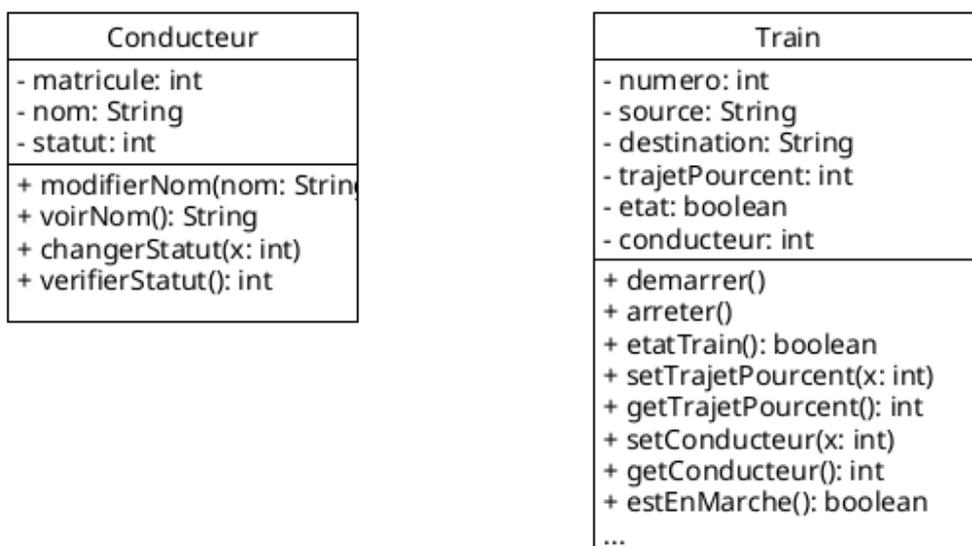
Le conducteur renseigne les passagers du train sur le pourcentage de trajet effectué (dans un système réel, ce serait fait par l'ordinateur de bord, mais ici, nous simplifions).

Les passagers du train pourront consulter l'avancement du trajet, mais pas le modifier.

Un train peut être à l'arrêt, ou en fonctionnement, c'est le conducteur qui gère cela.

A.3 Vision UML du cahier des charges

Les diagrammes de classe d'UML sont très pratiques pour voir les interactions entre les classes.



La représentation UML est pour le moment simplifié.

B Création de l'application

Dans l'éditeur Eclipse, vous allez créer un nouveau projet Java "Train" et vous allez créer un package "model" dans lequel nous mettrons les deux classes "Train" et "Conducteur".

B.1 Création du projet

Dans le menu **File** → **New** → **Project...** puis choisir **Java** → **Java Projects**

Saisir le nom du projet : **StationProject** et s'assurer que la case "Create module-info.java" est décochée.

Dans le gestionnaire de projet, sélectionnez StationProject et faire un clic droit **New** → **Package**

Nommez le package "**model**" et validez.

Sélectionnez maintenant votre package "model" et faites un clic droit, **New** → **Class** pour créer la classe "**Train**".

B.1.1 Classe Train

Complétez le code ci-dessous avec les éléments du cahier des charges. Essayez de compléter certaines méthodes vous-même.

```
package model;

public class Train {
    private Integer numero;
    private String source = "";
    private String destination = "";
    private String destination = "";
    // ajouter tous les attributs

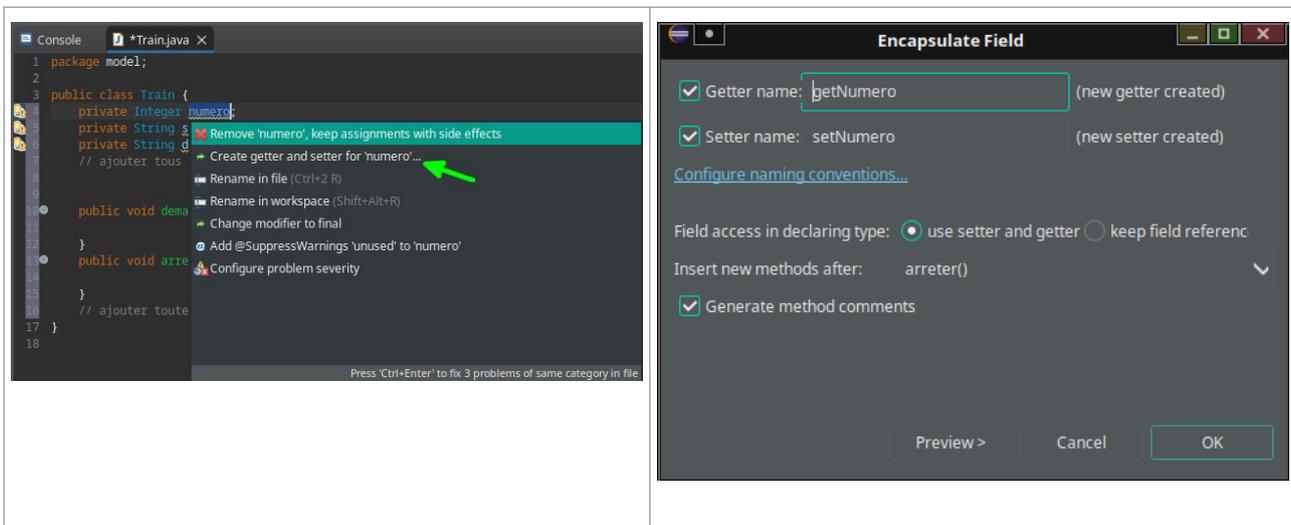
    public void demarrer() {
    }
    public void arreter() {
    }
    // ajouter toutes les méthodes
}
```

Nous allons également créer un constructeur pour instancier un train rapidement : les éléments importants sont le numéro du train, la ville source et la ville destination et le conducteur affecté au trajet.

Essayez également de

Vous pouvez notamment définir les accesseurs et les mutateurs. Un moyen simple pour gagner du temps, est de le faire faire par Eclipse : normalement, les attributs sont soulignés en jaune (pour indiquer qu'ils ne sont pas utilisés).

Un clic gauche sur le panneau attention à gauche des numéros de lignes affiche un menu d'aide permettant de choisir "Create getter and setter for 'numero'..." : Eclipse comprend alors qu'il doit fournir les méthodes que vous cocherez dans la fenêtre de dialogue qui apparaît.



Vous pouvez maintenant renommer les méthodes avec les noms choisis dans le diagramme UML.

Précisément, demarrer() et arreter() change l'attribut 'etat' à true ou false.

