

UML

Baptiste Bauer

Version v0.0.8.sip-221124100530, 2022-11-23 09:32:18

Table des matières

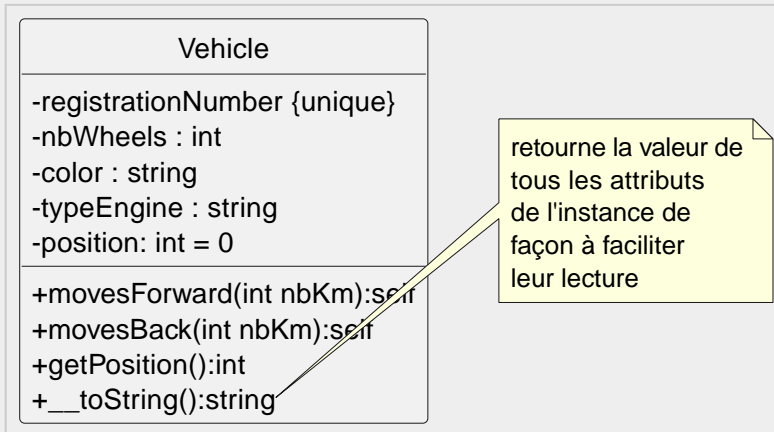
1. Quelques exercices d'implémentation	1
1.1. Implémenter une simple classe	1
1.2. Implémenter une association unidirectionnelle simple	1
1.3. Implémenter une association unidirectionnelle multiple avec cardinalité minimum à 0	2
1.4. Implémenter une association unidirectionnelle multiple avec cardinalité minimum à 1	3
1.5. Implémenter une association bidirectionnelle one-to-many	4

1. Quelques exercices d'implémentation

1.1. Implémenter une simple classe

Q1) Travail à faire

- Implémentez la classe suivante :



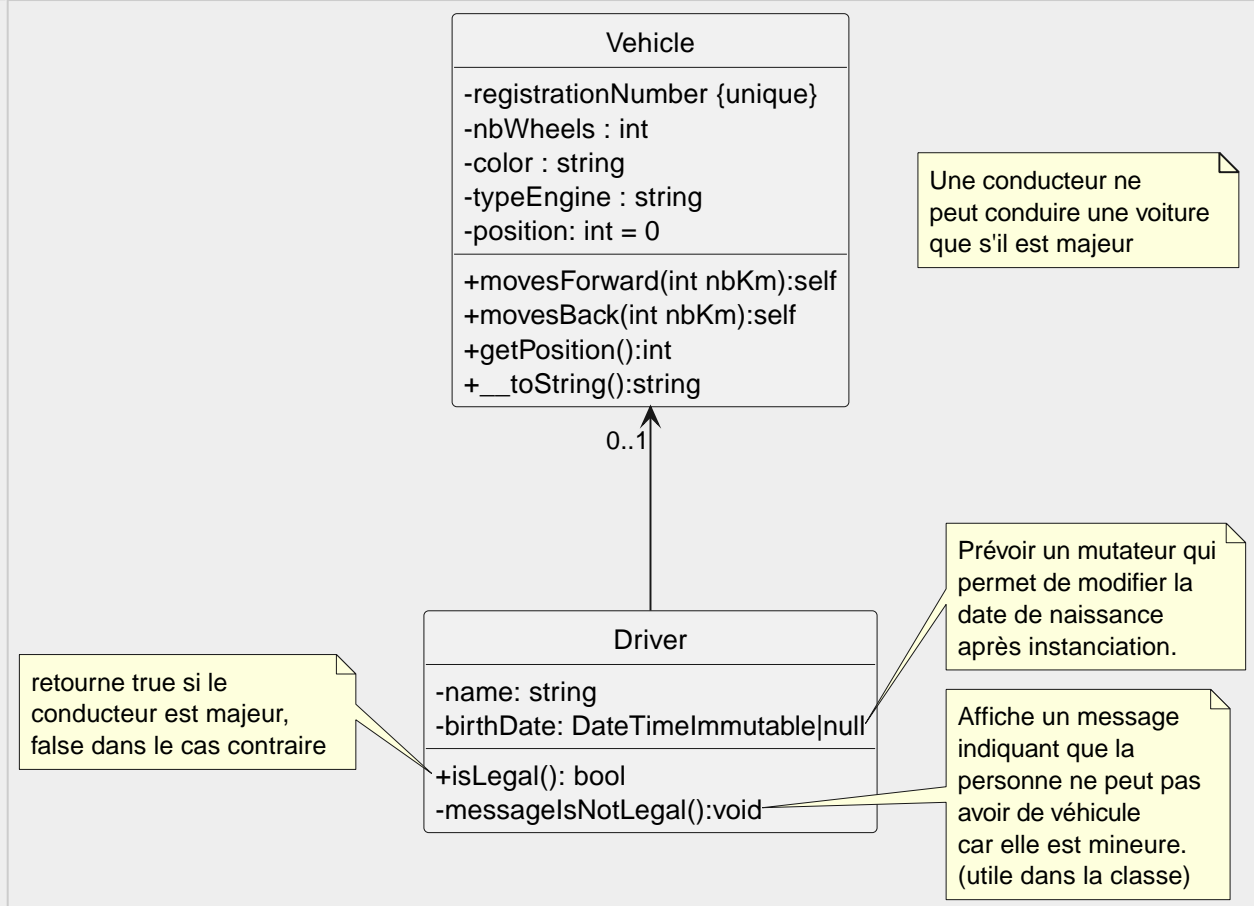
La position du véhicule ne doit pas pouvoir être affectée arbitrairement. Elle doit découler de la position initiale et de ses déplacements.

- Testez votre implémentation en créant deux véhicules différents et en les faisant avancer et reculer différemment, puis affichez les informations de chaque véhicule en mobilisant la méthode `__toString`.

1.2. Implémenter une association unidirectionnelle simple

Q2) Travail à faire

- Implémentez le diagramme suivant :



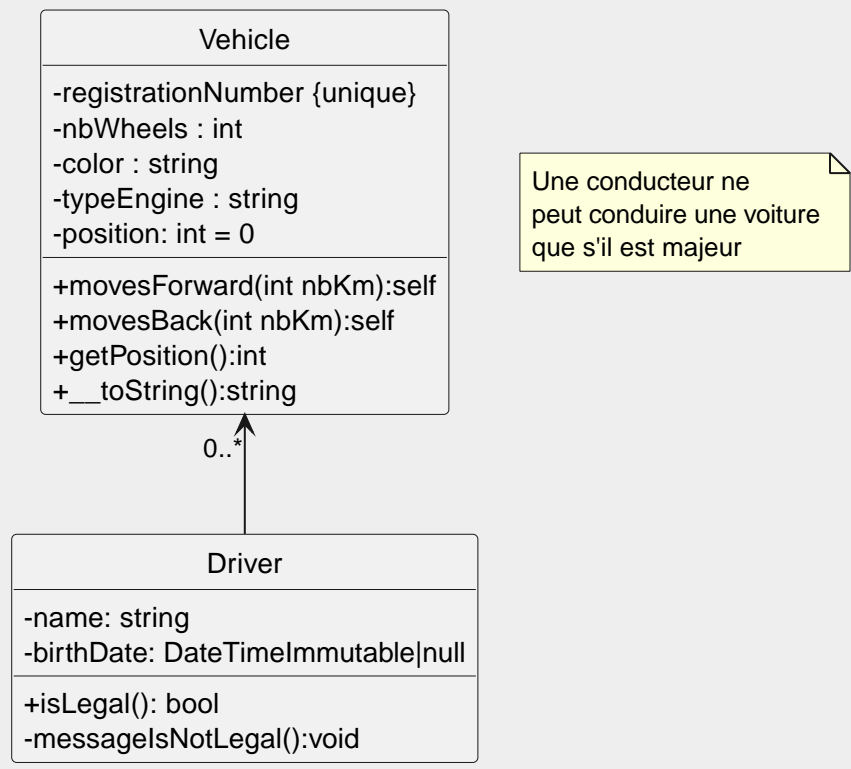
Si un véhicule est associé à un conducteur mineur, l'association ne doit pas se faire. Un message devra s'afficher indiquant que la personne est mineure et qu'il n'est pas possible d'associer un véhicule à un mineur. Attention à la situation qui consisterait à changer la date de naissance après que la personne se soit vue affecter un véhicule.

- Testez votre implémentation en essayant d'affecter un conducteur mineur à un premier véhicule et un autre conducteur à un second véhicule qui avance et recule selon votre bon vouloir.
- Donnez via le code la position du véhicule du conducteur majeur depuis la variable qui référence celui-ci.

1.3. Implémenter une association unidirectionnelle multiple avec cardinalité minimum à 0

Q3) Travail à faire

- Implémentez le diagramme suivant :

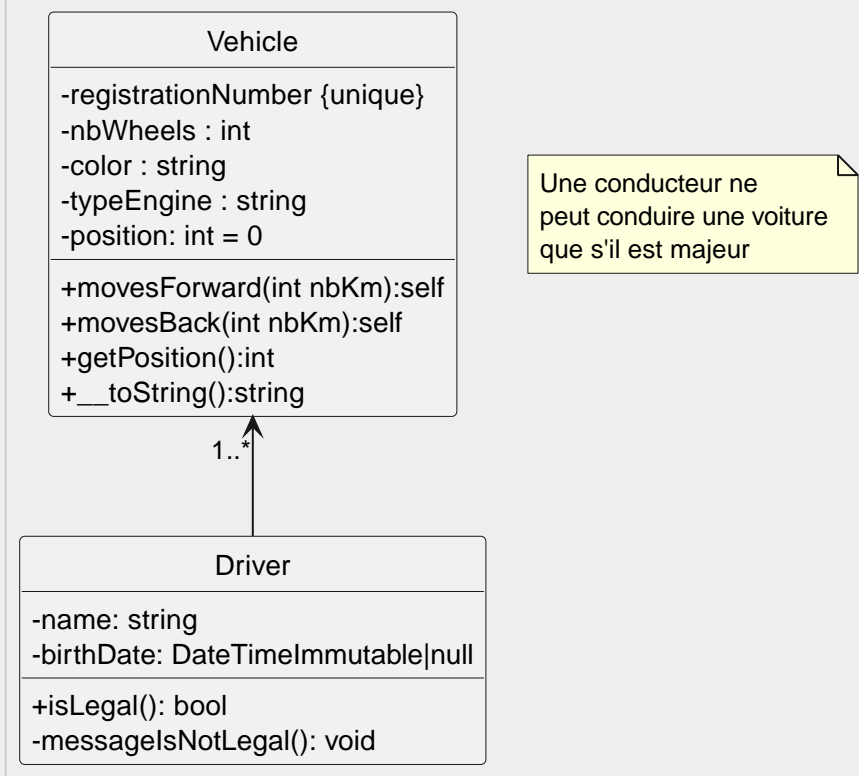


- Créez le conducteur Paul dont la date de naissance le conduit à être majeur
- Créez la conductrice Juliette dont la date de naissance la conduit à être mineure
- Créez les véhicules A, B, C, D et E avec les caractéristiques de votre choix
- Tentez d'affecter les véhicules A, C et E à Paul
- Tentez d'affecter les véhicules A et B à Juliette
- Faire avancer les véhicules A et C respectivement de 120km et 84km
- Faire reculer le véhicule C de 25km
- Afficher la liste des véhicules affectés à Paul
- Afficher la liste des véhicules affectés à Juliette
- Retirez les véhicules A et E à Paul
- Afficher la liste des véhicules restant à Paul

1.4. Implémenter une association unidirectionnelle multiple avec cardinalité minimum à 1

Q4) Travail à faire

- Implémentez le diagramme suivant :

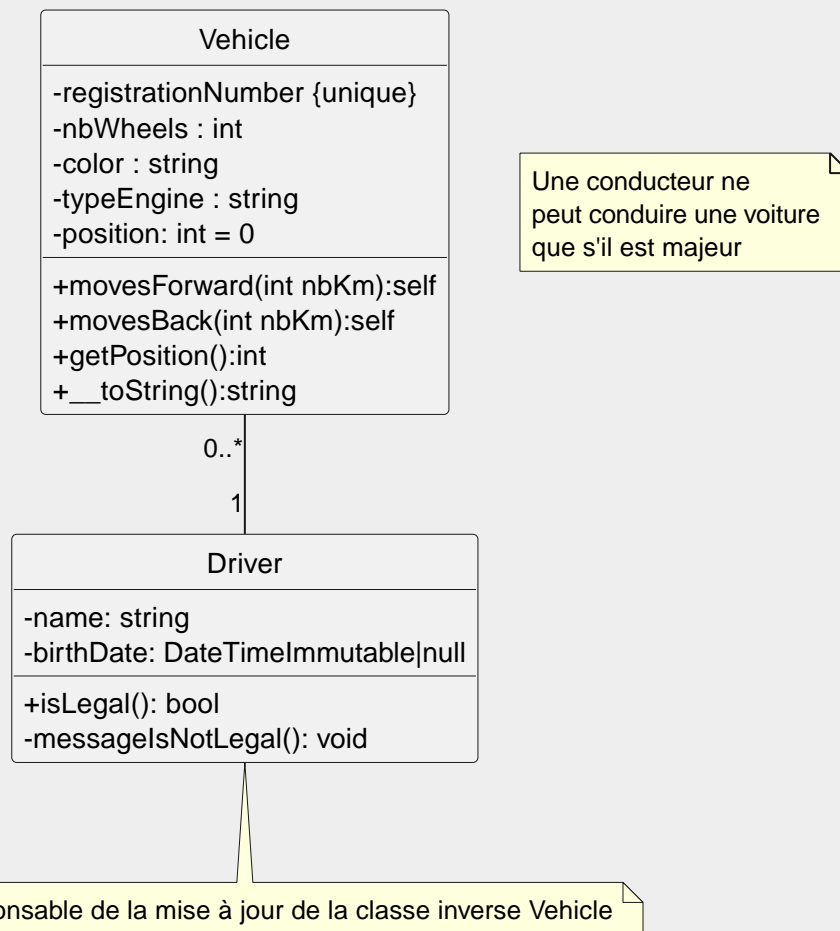


- Créez le conducteur Paul dont la date de naissance le conduit à être majeur. Il conduira le véhicule A.
- Affectez un véhicule supplémentaire B à Paul
- Créez un véhicule C (sans conducteur)
- Faire avancer les véhicules A et C respectivement de 123km et 257km
- Faire reculer le véhicule A de 70km
- Afficher depuis la variable référençant Paul la liste de ses véhicules et leur position.
- Retirez les véhicules A et C à Paul
- Afficher depuis la variable référençant Paul la liste de ses véhicules et leur position.

1.5. Implémenter une association bidirectionnelle one-to-many

Q5) Travail à faire

- Implémentez le diagramme suivant :



- Créez le conducteur Paul dont la date de naissance le conduit à être majeur.
- Créez le conducteur Juliette dont la date de naissance la conduit à être mineure.
- Créez un véhicule A et tenter de lui affecter Juliette.
- Créez un véhicule B et tenter de lui affecter Paul.
- Afficher la liste des véhicules associés à Paul.
- Afficher la liste des véhicules associés à Juliette.