UML

Baptiste Bauer

Version v0.0.8.sip-221124091855, 2022-11-23 09:32:18

Table des matières

1. L'interface

1.1. Notion d'interface et modélisation UML

Il faut avoir parfaitement compris ce qu'est une relation abstraite pour comprendre la relation d'interface.

Une interface est une **classe dont toutes les méthodes sont publiques et abstraites**.

Cela ressemble à une classe abstraite sauf que contrairement à elle, une interface n'a que des méthodes sans corps. Il n'y a donc que la signature d'une ou plusieurs méthodes.

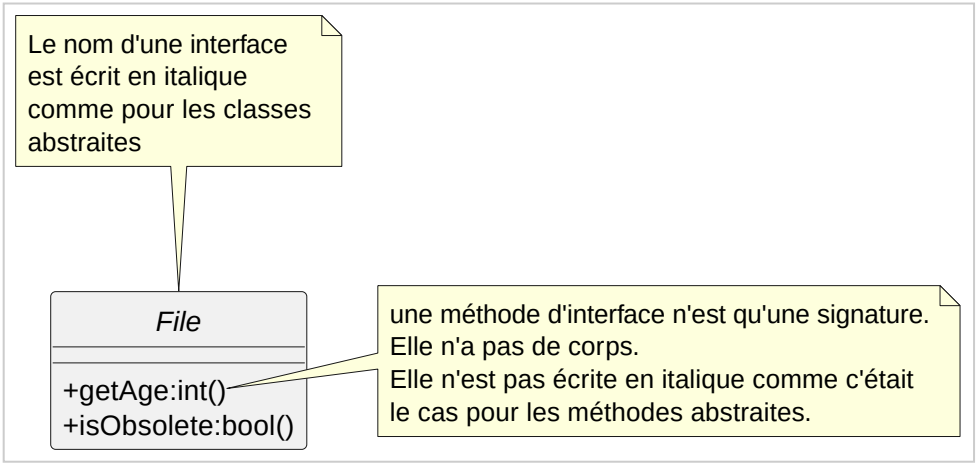
Vous devez vous demander à quoi peut servir une "classe" qui n'a que des méthodes abstraites.

Dans la partie sur la relation abstraite, nous avons vu qu'une classe qui hérite d'une classe abstraite qui contient des méthodes abstraites doit obligatoirement implémenter celles-ci. La classe qui hérite est contrainte, elle n'a pas le choix, elle DOIT implémenter ces méthodes.

Donc, si une classe *hérite* d'une interface, elle devra obligatoirement implémenter les méthodes de celle-ci. C'est très utile pour contraindre les classes à être manipulées avec des méthodes qui sont prévisibles car définies par l'interface.

En programmation, on ne dit pas qu'une classe *hérite* d'une interface, on dit qu'une classe **implémente une interface**.

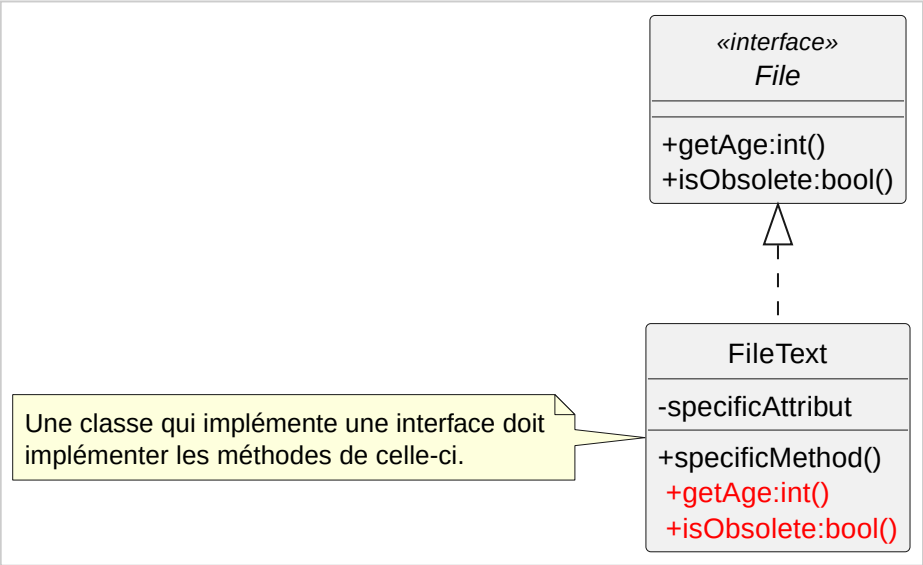
Le mieux est d'illustrer cette logique par un diagramme :



Il est difficile de distinguer une interface d'une classe abstraite (dans les deux cas, le nom est écrit en italique). Il est possible d'utiliser le stéréotype pour une meilleure lisibilité :

Nous avons dans ce diagramme une interface (il ne faut pas utiliser le terme de "classe" dans le cadre d'une interface) qui prévoit deux méthodes. Cela signifie que toute classe implémentant cette interface devra prévoir l'implémentation de ces 2 méthodes.

Ajoutons une classe `FileText` qui **implémente** l'interface `File` :



1.2. Implémentation d'une relation avec une interface

Une interface est déclarée (en PHP) avec le mot `interface` :

```
1 interface File ①
2 {
3     /**
4      * @return int retourne la taille du fichier en Mo
5      */
6     public function getSize():int; ②
7
8     /**
9      * @return int retourne l'âge du fichier en mois
10     */
11    public function getAge():int; ②
12
13 }
```

① Déclaration d'une interface

② Signature des méthodes qui devront être implémentées par les classes qui vont implémenter l'interface

Maintenant que notre interface est en place, nous pouvons indiquer à la classe `FileText` de l'implémenter :

```
1 class FileText implements File{ ①
2
```

```
3     //ici les membres spécifiques de FileText
4
5 }
```

① La classe **implémente** l'interface `File` .

Enfin, il ne reste plus qu'à implémenter les méthodes de l'interface :

```
1 class FileText implements File{ ①
2
3     //ici les membres spécifiques de FileText
4
5
6     public function getAge(): int
7     {
8         // ici du code qui retourne un nombre de mois
9     }
10
11    public function getSize(): int
12    {
13        // ici du code qui retourne un nombre de Mo
14    }
15 }
```

① Notre classe implémente toutes les méthodes de l'interface

Si une classe ne peut pas hériter de plusieurs classes mères, elle peut implémenter plusieurs interfaces :

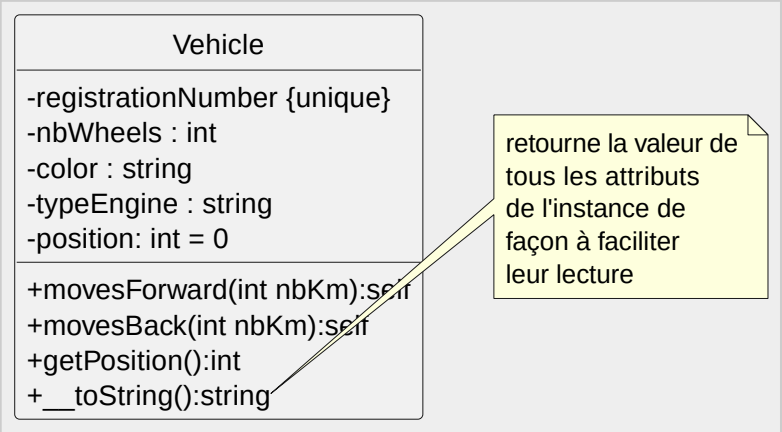
```
1 class classA implements classB, classC, classD {
2     //ici les membres spécifiques de classA
3
4     //ici toutes les méthodes déclarées dans les interfaces classB,
  classC et classD
5 }
```

2. Quelques exercices d'implémentation

2.1. Implémenter une simple classe

Q1) Travail à faire

- Implémentez la classe suivante :



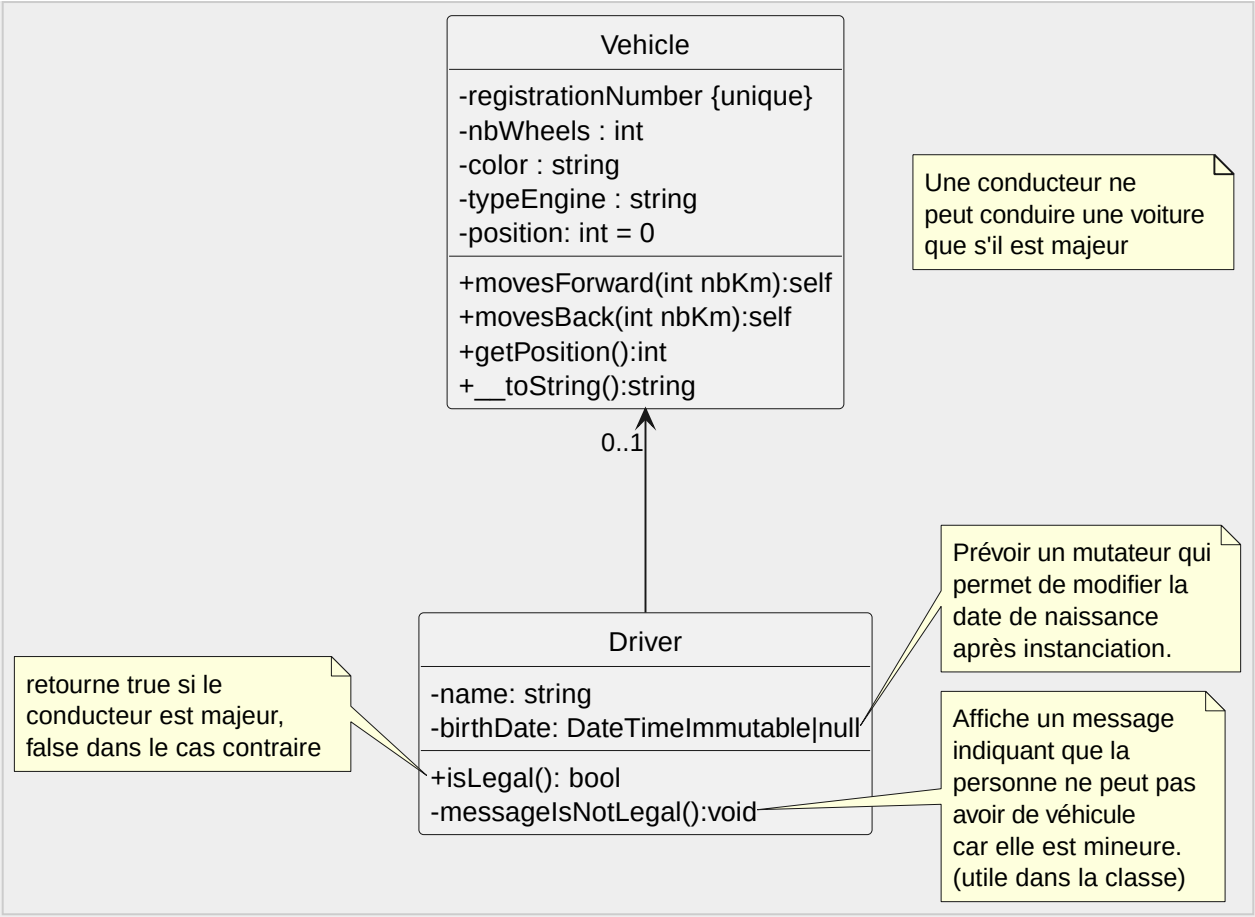
La position du véhicule ne doit pas pouvoir être affectée arbitrairement. Elle doit découler de la position intiale et de ses déplacements.

- 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`.

2.2. Implémenter une association unidirectionnelle simple

Q2) Travail à faire

- Implémentez le diagramme suivant :



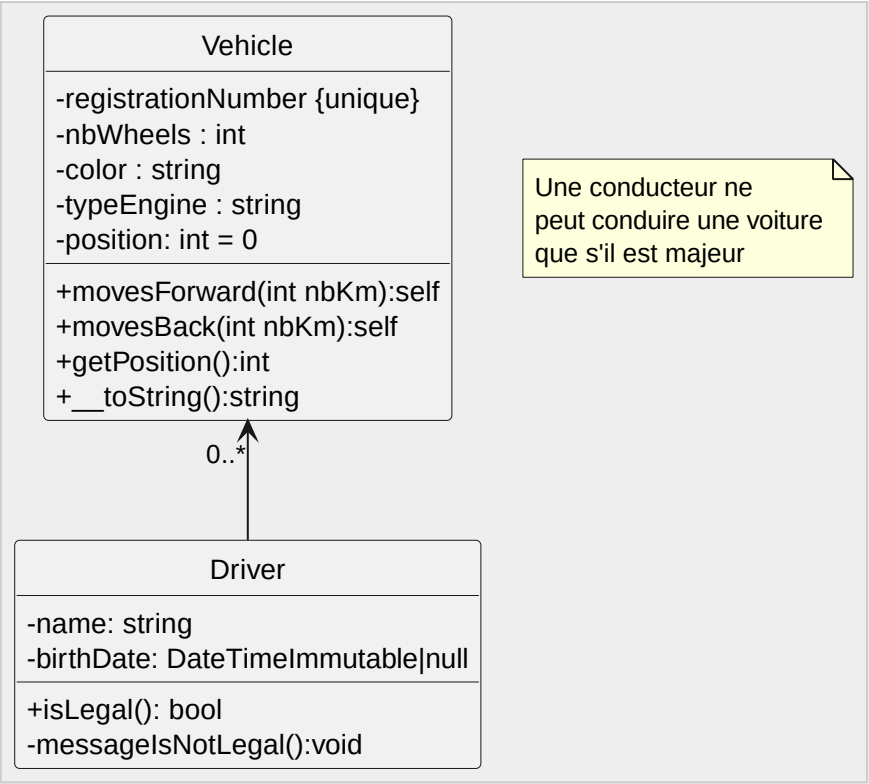
Si un véhicule est associé à un conducteur mineur, l'association ne doit pas se faire. Un message devra s'afficher indiquant que la personne est mineure et qu'il n'est pas possible d'associer un véhicule à un mineur. Attention à la situation qui consisterait à changer la date de naissance après que la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

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

Q3) Travail à faire

- Implémentez le diagramme suivant :

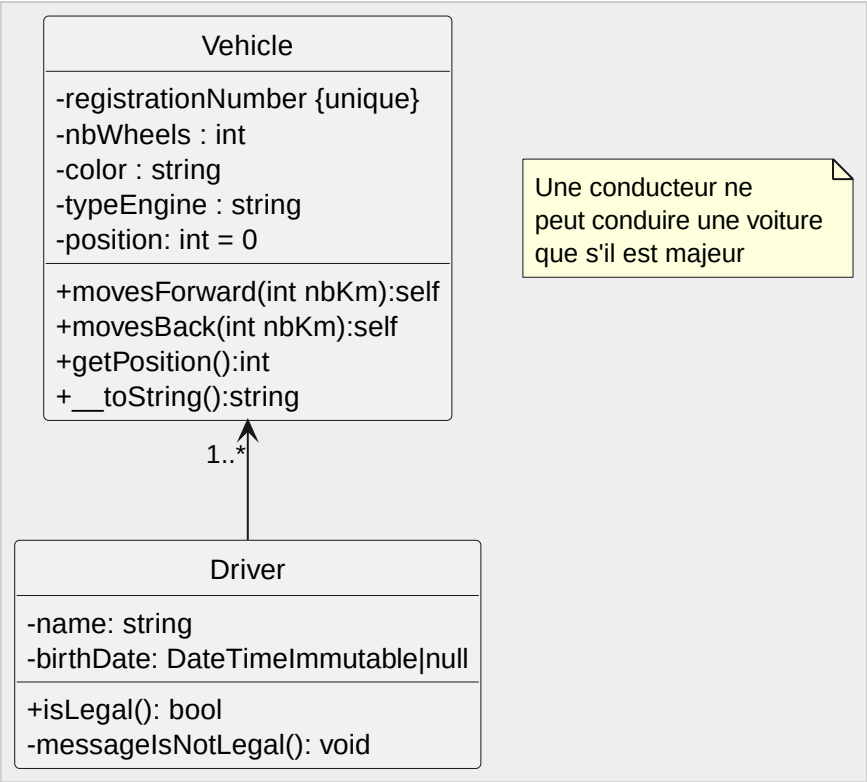


- Créez le conducteur Paul dont la date de naissance le conduit à être majeur
- Créez la conductrice Juliette dont la date de naissance la conduit à être mineure
- Créez les véhicules A, B, C, D et E avec les caractéristiques de votre choix
- Tentez d'affecter les véhicules A, C et E à Paul
- Tentez 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

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

Q4) Travail à faire

- Implémentez le diagramme suivant :

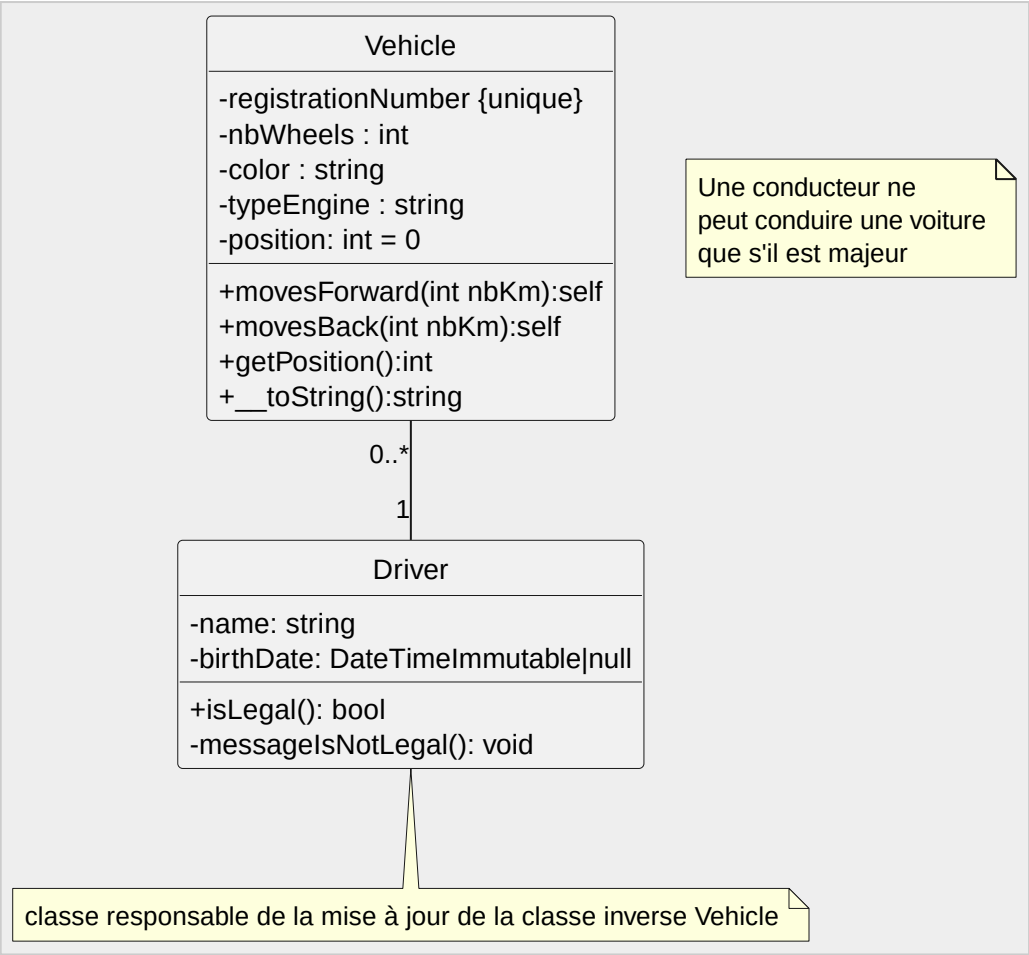


- Créez le conducteur Paul dont la date de naissance le conduit à être majeur. Il conduira le véhicule A.
- Affectez un véhicule supplémentaire B à Paul
- Créez un véhicule C (sans conducteur)
- Faire avancer les véhicules A et C respectivement de 123km et 257km
- Faire reculer le véhicule A de 70km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

2.5. 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.

Index