

UE Conception Orientée Objet

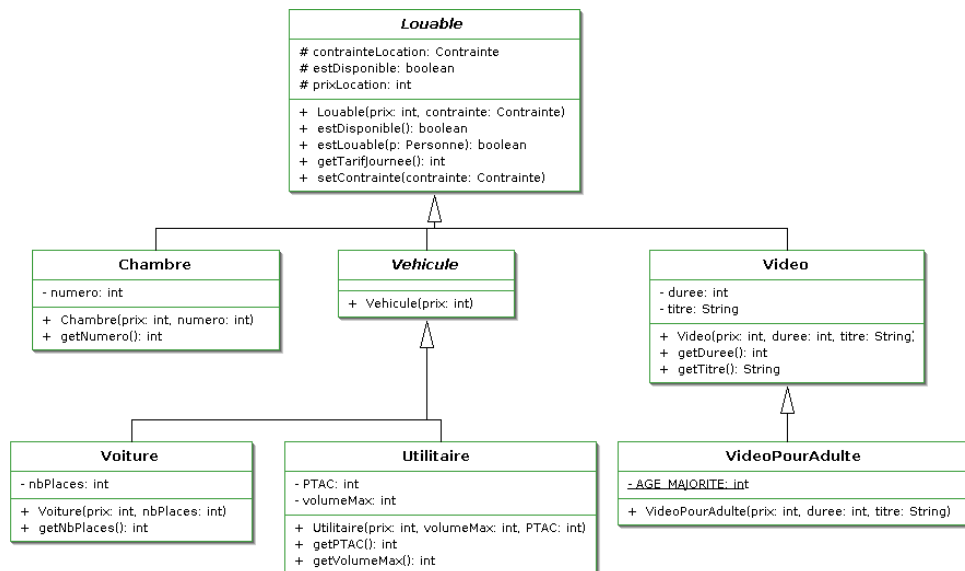
Contraintes de location

On s'intéresse à des "choses" qu'il est possible de louer à la journée. Pour être louée une telle chose, de type **Louable**, doit être disponible (méthode `estDisponible() :boolean`) et le client-locataire (de type **Personne**) doit satisfaire la contrainte de location (méthode `estLouable(p : Personne) :boolean`). Un prix de location à la journée, fixée à la construction, est associé à chaque objet (méthode `getTarifJournee() :float`).

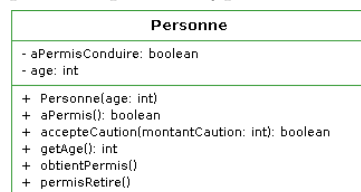
Parmi les "choses" louables on trouve :

- les *vidéos*, qui sont caractérisées par une durée et un titre, il n'y a par défaut pas de contrainte de location pour les vidéos, cependant on distingue le cas particulier des "vidéos pour adultes" que seul un majeur peut louer ;
- les *chambres* (louées dans un hôtel), qui disposent d'un numéro (méthode `getNumero() :int`), la contrainte de location est une caution égale au tarif de la location pour une journée ;
- les *véhicules* pour lesquels la contrainte de location est d'avoir le permis. Parmi les véhicules on distingue les *voitures* qui sont caractérisées par un nombre de passagers possible (méthode `getNbPlaces() :int`) et les *véhicules utilitaires* qui sont caractérisés par un volume (méthode `getVolumeMax() :float`) et un poids maximum transporté `getPTAC() :int`.

Voici par exemple un diagramme UML pour de telles entités :



Et maintenant un diagramme UML possible pour le type **Personne** :



Contraintes. Les contraintes de location qui s'appliquent aux objets louables peuvent être de natures différentes :

- *aucune contrainte* : tout le monde peut louer
- *contrainte d'âge* : il faut avoir un âge minimum fixé par la contrainte pour louer l'objet
- *contrainte permis* : il faut avoir le permis de conduire pour louer l'objet
- *contrainte caution* : il faut que le client accepte de laisser une caution d'un montant fixé à la construction de la contrainte
- etc.

Les contraintes sont modélisées par un type **Contrainte** qui dispose d'une méthode `estSatisfaite(p :Personne) :boolean` dont le résultat est `true` si la personne `p` satisfait la contrainte (`p` peut donc louer l'objet concerné).

Il existe également des *contraintes multiples* qui cumulent les conditions de plusieurs autres contraintes, par exemple elles imposent qu'il est nécessaire d'avoir le permis et de déposer une caution, ou encore qu'il faut avoir le permis et au moins 21 ans.

- Q 1 .** Indiquez sous forme de diagrammes UML comment vous proposez de modéliser les contraintes, en faisant apparaître celles mentionnées dans ce sujet. En particulier, faites apparaître clairement votre proposition pour les contraintes *multiples* .
- Q 2 .** Donnez le code java de la méthode `estSatisfaite` pour les contraintes multiples.