

Introduction à la modélisation conceptuelle de données avec UML

Attribution - Partage dans les Mêmes Conditions :
<http://creativecommons.org/licenses/by-sa/3.0/fr/>

Table des matières

I - Contexte	3
II - Le formalisme UML	4
III - Exercice	6
IV - Classes	7
V - Exercice	9
VI - Attributs, types et méthodes	10
VII - Exercice	13
VIII - Associations	14
IX - Exercice	17
X - Classe d'association	18
XI - Exercice	20
XII - Unicité, immuabilité, optionnalité	21
XIII - Exercice	23
XIV - Élaboration d'un diagramme UML	24
XV - Exercice	27
XVI - Documentation avec PlantUML	28
XVII - Exercice	31
XVIII - Essentiel	32
XIX - Quiz	33
Solutions des exercices	37
Abréviations	46
Index	47

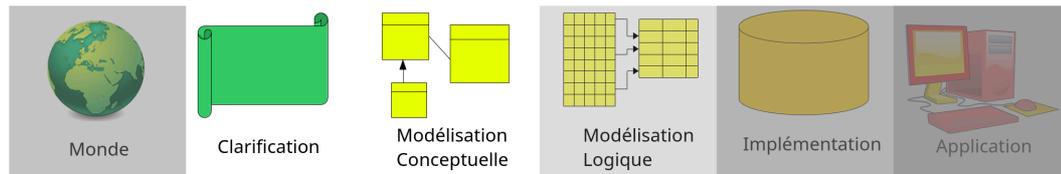
I Contexte

Durée : 3h

Environnement de travail : PlantUML

Pré-requis : Aucun

Après la première étape de clarification de la situation dans le processus de conception qui vise à expliciter les détails sur le système à concevoir, on réalise une étape de modélisation conceptuelle de cette situation.



Étape de modélisation conceptuelle

La modélisation conceptuelle est l'étape fondatrice du processus de conception de BD. Elle consiste à abstraire le problème réel posé pour en faire une reformulation qui trouvera une solution dans le cadre technologique d'un SGBD. Si le modèle dominant en conception de bases de données a longtemps été le modèle E-A, le modèle UML se généralise de plus en plus. Parmi les divers diagrammes proposés par le standard UML, en conception de base de données on utilise essentiellement le diagramme de classe.

Nous proposons ici une introduction au diagramme de classes à travers la représentation de classes et d'associations simples.

II Le formalisme UML

Objectifs

- Comprendre la notion d'objet ;
- Connaître les différents diagrammes UML.

Mise en situation

Que ce soit pour décrire un processus, comme l'échange de données entre un navigateur web et un serveur, ou pour décrire une situation réelle, comme un ensemble de personnes, il est nécessaire de passer par une modélisation.

La modélisation d'un système, quel qu'il soit, peut se faire de différentes façons : par exemple en expliquant avec des phrases ou en dessinant un schéma.

Quand on utilise un schéma, l'idéal c'est que tout le monde puisse le comprendre, qu'il utilise une sorte de langage universel pour décrire n'importe quelle situation.

C'est l'objectif d'UML, qui permet de modéliser graphiquement une grande variété de situations.

Méthode

UML propose un formalisme qui impose de « penser objet » et permet de rester indépendant d'un langage de programmation donné lors des premières phases de conception.

UML permet :

- d'énumérer et de décrire les concepts,
- de les noter graphiquement..

Il est utilisé comme un moyen de communication entre les étapes de spécification conceptuelle et les étapes de spécifications techniques.

UML ^{p.46} est un langage de représentation destiné en particulier à la modélisation objet. UML est devenu une norme *OMG* ^{p.46} en 1997.

Dans le domaine des bases de données UML peut être utilisé pour réaliser l'étape de modélisation conceptuelle.

Remarque

Il existe plusieurs diagrammes au sein du formalisme UML, dont les principaux sont :

- les diagrammes de classes, qui permettent de décrire une situation dans un formalisme de classes,
- les diagrammes de séquences, qui décrivent les différentes interactions entre les éléments d'un logiciel suite à une sollicitation

Méthode

En BD on utilise le **diagramme de classe** d'UML pour modéliser conceptuellement les données.

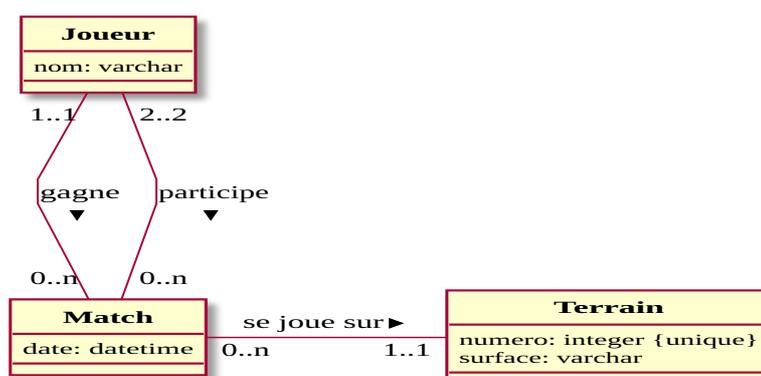
Diagramme de classe

Fondamental

Le diagramme de classes est un sous ensemble d'UML qui s'attache à la description statique d'un modèle de données représentées par des classes d'objets.

La modélisation d'un match de tennis avec un diagramme de classes UML

Exemple



Match de tennis – Diagramme de classe

Rappel

Un schéma conceptuel UML sera traduit en schéma logique relationnel à la seconde étape du processus de conception d'une base de données relationnelle.

Exercice

[solution n°1 p. 37]

Soit le modèle conceptuel suivant représentant des visites dans un centre médical. Quelles sont les assertions vraies selon ce schéma ?

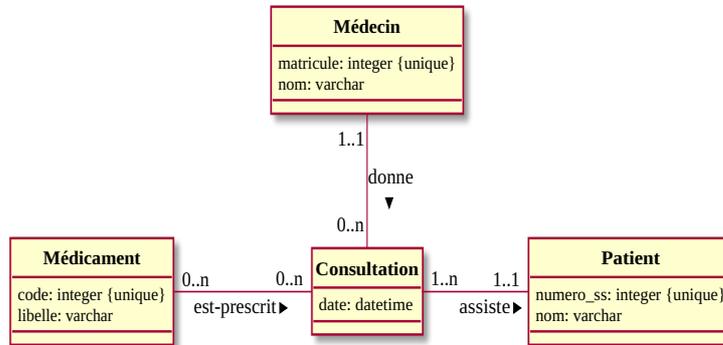


Image 1

- A** Un patient peut effectuer plusieurs visites.
- B** Tous les patients ont effectué au moins une consultation.
- C** Un médecin peut recevoir plusieurs patients pendant la même consultation.
- D** Un médecin peut prescrire plusieurs médicaments lors d'une même consultation.
- E** Deux médecins différents peuvent prescrire le même médicament.

IV Classes

Objectif

- Comprendre la notion de classe en UML.

Mise en situation

Pensez à n'importe quel objet, au sens large, de la réalité : une assiette, un chat, un livre, une page web, une carotte, etc.

Maintenant, essayez de réfléchir à ce qui caractérise cet objet, et plus particulièrement, ce qui différencie une assiette d'une autre, ou un chat d'un autre. Quelques idées viennent en tête : la race, la couleur, le poids, le nom, etc.

Toutes ces caractéristiques forment un modèle de l'objet que vous décrivez, et dans ce module, vous allez apprendre à utiliser les classes UML pour regrouper ces caractéristiques et définir une sorte de squelette permettant de décrire tous les chats, ou toutes les carottes.

Classe

Az Définition

Une classe est un type abstrait caractérisé par des propriétés (attributs et méthodes) communes à un ensemble d'objets et permettant de créer des instances de ces objets, ayant ces propriétés.

Syntaxe

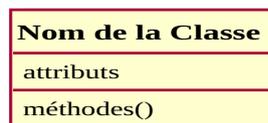


Image 2

Exemple



Image 3

[👁 Exemple](#)

```
Voiture  
marque: varchar  
type: varchar  
nb_portes: integer  
puissance: integer  
kilométrage: integer
```

Image 4

Une instance de la classe Voiture

[👁 Exemple](#)

L'objet V1 est une instance de la classe Voiture.

V1 : Voiture

- **Marque** : 'Citroën'
- **Type** : 'Visa'
- **Portes** : 5
- **Puissance** : 60
- **Kilométrage** : 300000

[⊕ Complément](#)

La modélisation sous forme de diagramme de classes est une modélisation statique, qui met en exergue la structure d'un modèle, mais ne rend pas compte de son évolution temporelle. UML propose d'autres types de diagrammes pour traiter, notamment, de ces aspects.

V Exercice

Question

[solution n°2 p. 37]

Un instrument de musique est rattaché à un modèle, a été fabriqué par un fabricant, dispose d'un numéro de série, a une certaine couleur et a été fabriqué avec un matériau principal.

Représentez un diagramme UML d'une classe composée de 5 attributs.

On utilisera papier et crayon et on considérera pour simplification que les attributs sont tous de type `varchar` associés aux chaînes de caractères.

VI Attributs, types et méthodes

Objectifs

- Comprendre ce qu'est un attribut au sens UML ;
- Découvrir le typage des attributs.

Mise en situation

Toute modélisation d'un objet de la réalité se décompose en petits bouts élémentaires : pour une personne, on pourra penser à son nom, son prénom, sa date de naissance, son âge, etc.

En UML, ces petits bouts sont appelés attributs.

Or, chaque attribut a ses spécificités : on peut faire des additions avec l'âge, on peut mettre un nom en majuscule, etc. En revanche, on ne peut pas additionner deux prénoms. Comment différencier la nature des attributs, et ce que l'on peut faire avec ?

Dans ce module, vous allez apprendre à caractériser les attributs des classes UML grâce aux types et aux méthodes.

Attribut

Az Définition

Un attribut est une information élémentaire qui caractérise une classe et dont la valeur dépend de l'objet instancié.

Un attribut est typé : Le domaine des valeurs que peut prendre l'attribut est fixé *a priori*.

Exemple

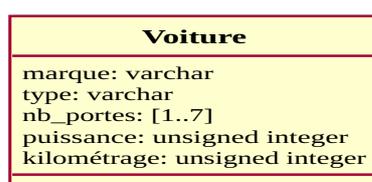


Image 5

Types

Syntaxe

Le typage des attributs peut se faire dans une syntaxe libre, il n'est exprimé que pour :

- consigner des contraintes,
- expliciter la nature de l'attribut.

On peut utiliser une terminologie française, anglaise, inventée, liée à un langage, etc. On veillera néanmoins à conserver une terminologie cohérente.

Types en SQL

 Syntaxe

On peut utiliser sinon les types de SQL, voici les plus courants :

- VARCHAR pour des chaînes de caractères à longueur variable ;
- TEXT pour de long texte ;
- INTEGER pour les entiers ;
- DECIMAL pour les nombres décimaux à partie entière et flottante connue ;
- REAL pour les nombres décimaux sans restriction sur les parties entière et flottante ;
- DATE pour les dates ;
- DATETIME pour les horodatages ;
- TIME pour les heures ;
- BOOLEAN pour les booléens.

Précision des types

 Conseil

On conseille d'adopter sur le schéma un typage précis qui rend compte de ce qui est exprimé dans l'analyse des besoins. Si l'on sait qu'une voiture ne peut pas avoir plus de sept portes, on le consigne dès l'UML dans le typage.

Méthode

Az Définition

Une méthode est une fonction associée à une classe qui permet d'agir sur les objets de la classe ou qui permet à ces objets de renvoyer des valeurs.

Une méthode peut donc être vue comme un attribut dans la valeur est calculée et non stockée.

 Syntaxe

Les méthodes se notent sous les attributs.

Voiture
marque: varchar
type: varchar
nb_portes: [1..7]
puissance: unsigned integer
kilométrage: unsigned integer
mise_en_circulation: date
age(): integer

Méthodes et modélisation de BD

 Méthode

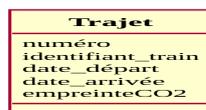
Pour la modélisation des bases de données, les méthodes sont utilisées pour représenter des données calculées.

VII Exercice

Un trajet en train est composé de plusieurs informations :

- un numéro,
- un identifiant de train,
- une date de départ,
- une date d'arrivée,
- une empreinte CO2.

Ces informations sont résumées dans le diagramme de classe suivant :



Question 1

[solution n°3 p. 38]

Compléter ce diagramme pour y ajouter les types. Bien que la dénomination des types soit libre, on préférera utiliser les types de SQL.

Question 2

[solution n°4 p. 38]

Grâce aux dates, on peut calculer la durée du trajet en nombre de minutes.

Ajouter la méthode correspondante sur le diagramme avec un type adapté.

VIII Associations

Objectifs

- Savoir représenter les relations entre des classes grâce aux associations ;
- Savoir ajouter des contraintes sur les associations.

Mise en situation

Imaginez que vous développez une application à destination des agences immobilières pour faciliter la gestion des locations d'appartements ou de maisons.

Dans votre modélisation, vous avez identifié plusieurs acteurs : les locataires, les bailleurs, les biens immobiliers, etc. Chacun de ces acteurs est représenté par une classe UML.

Mais comment représenter les relations qui existent entre ces classes ? Par exemple, un bailleur peut posséder plusieurs biens immobiliers, mais un bien immobilier ne peut être loué que par un seul locataire.

Dans ce module, vous allez découvrir les associations UML, qui permettent d'exprimer efficacement les relations entre plusieurs classes, ainsi que leurs contraintes.

Association

Az Définition

Une association est une relation logique entre deux classes (association binaire) ou plus (association n-aire) qui définit un ensemble de liens entre les objets de ces classes.

Une association est **nommée**, généralement par un verbe. Une association peut avoir des propriétés (à l'instar d'une classe). Une association définit le nombre minimum et maximum d'instances autorisées dans la relation (on parle de cardinalité).

Syntaxe



Image 6

Attention

Le nom de l'association (verbe qui la décrit) est obligatoire, au même titre que le nom d'une classe ou d'un attribut.

Remarque

Une association est généralement bidirectionnelle (c'est à dire qu'elle peut se lire dans les deux sens). Les associations qui ne respectent pas cette propriété sont dites unidirectionnelles ou à navigation restreinte.

L'association Conduit

Exemple

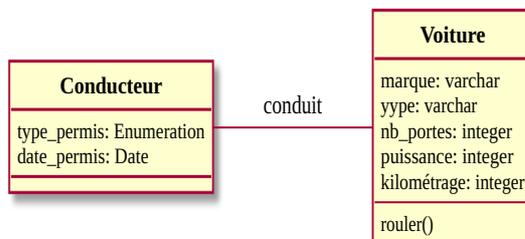


Image 7

L'association « *Conduit* » entre les classes « *Conducteur* » et « *Voiture* » exprime que les conducteurs conduisent des voitures.

Cardinalité d'une association

Az Définition

La cardinalité d'une association permet de représenter le nombre minimum et maximum d'instances qui sont autorisées à participer à la relation. La cardinalité est définie pour les deux sens de la relation.

Syntaxe

Si \min_a (resp. \max_a) est le nombre minimum (resp. maximum) d'instances de la classe A autorisées à participer à l'association, on note sur la relation, à côté de la classe A : $\min_a..max_a$.

Si le nombre maximum est indéterminé, on note n ou *.

Remarque

Les cardinalités les plus courantes sont :

- 0..1 (optionnel)
- 1..1 ou 1 (un)
- 0..n ou 0..* ou * (plusieurs)
- 1..n ou 1..* (obligatoire)

La cardinalité de l'association Possède

👁 Exemple

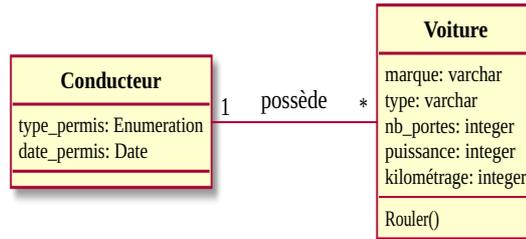


Image 8

Ici un conducteur peut posséder plusieurs voitures (y compris aucune) et une voiture n'est possédée que par un seul conducteur.

Terminologie

⚠ Attention

- On appelle association 1:1 les associations de type :
 - 0..1:0..1
 - 0..1:1..1
 - 1..1:0..1
 - 1..1:1..1
- On appelle association 1:N les associations de type :
 - 0..1:0..N
 - 0..1:1..N
 - 1..1:0..N
 - 1..1:1..N
- On appelle association N:M (ou M:N) les associations de type :
 - 0..N:0..M
 - 0..N:1..M
 - 1..N:0..M
 - 1..N:1..M

⊕ Complément

La notation de la cardinalité en UML est opposée à celle adoptée en E-A. En UML on note à gauche (resp. à droite) le nombre d'instances de la classe de gauche (resp. de droite) autorisées dans l'association. En E-A, on note à gauche (resp. à droite) le nombre d'instances de la classe de droite (resp. de gauche) autorisées dans l'association.

IX Exercice

Description du problème

Un laboratoire souhaite gérer les médicaments qu'il conçoit.

- Un médicament est décrit par un nom, qui permet de l'identifier. En effet il n'existe pas deux médicaments avec le même nom. Un médicament comporte une description courte en français, ainsi qu'une description longue en latin. On gère aussi le conditionnement du médicament, c'est à dire le nombre de pilules par boîte (qui est un nombre entier).
- À chaque médicament on associe une liste de contre-indications, généralement plusieurs, parfois aucune. Une contre-indication comporte un code unique qui l'identifie, ainsi qu'une description. Une contre-indication est toujours associée à un et un seul médicament.

Exemple de données

Afin de matérialiser notre base de données, nous obtenons les descriptions suivantes :

- Le **Chourix** a pour description courte « *Médicament contre la chute des choux* » et pour description longue « *Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae, sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare.* ». Il est conditionné en boîte de 13.

Ses contre-indications sont :

- CI1 : Ne jamais prendre après minuit.
- CI2 : Ne jamais mettre en contact avec de l'eau.

- Le **Tropas** a pour description courte « *Médicament contre les dysfonctionnements intellectuels* » et pour description longue « *Suspendisse lectus leo, consectetur in tempor sit amet, placerat quis neque. Etiam luctus porttitor lorem, sed suscipit est rutrum non.* ». Il est conditionné en boîte de 42.

Ses contre-indications sont :

- CI3 : Garder à l'abri de la lumière du soleil.

Question 1

[solution n°5 p. 38]

Réaliser le modèle conceptuel de données en UML du problème.

Question 2

[solution n°6 p. 38]

Étendre le modèle conceptuel UML afin d'ajouter la gestion des composants. Un composant est identifié par un code unique et possède un intitulé. Tout médicament possède au moins un composant, souvent plusieurs. Tout composant peut intervenir dans la fabrication de plusieurs médicaments. Il existe des composants qui ne sont pas utilisés pour fabriquer des médicaments et que l'on veut quand même gérer.

X Classe d'association

Objectifs

- Savoir ajouter des propriétés à une association ;
- Connaître les cas d'utilisation des classes d'association.

Mise en situation

Imaginez que vous développez une application de vente en ligne, où les clients peuvent passer des commandes. Vous avez identifié plusieurs classes UML, comme par exemple les articles et les clients. Vous avez également identifié une association entre client et article, qui symbolise l'action de commander un article.

Mais dans votre modélisation, si un client veut commander plusieurs articles, où va aller cette information ? Elle n'est pas directement un attribut du client, ni de l'article ?

C'est plutôt un attribut de l'association, lié à l'acte de commande. Dans ce module, vous allez apprendre à ajouter des attributs aux associations, grâce aux classes d'association.

Classe d'association

Az Définition

On utilise la notation des classes d'association lorsque l'on souhaite ajouter des propriétés à une association.

Notation d'une classe d'association en UML

Syntaxe

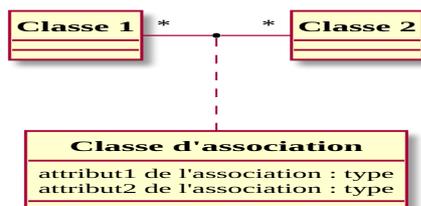


Image 9

Méthode

On réserve en général les classes d'association aux associations N:M.

En effet, il est toujours possible de réduire une classe d'association sur une association 1:N en migrant ses attributs sur la classe côté N, et c'est en général plus lisible ainsi.

Exemple de classe d'association

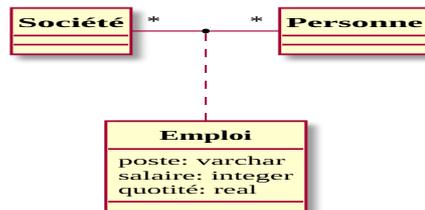
[Exemple](#)

Image 10

[Conseil](#)

Selon le standard UML une classe d'association est une classe et à ce titre elle peut être mobilisée dans d'autres associations ou dans des héritages. Nous déconseillons néanmoins ces notations qui ont tendance à complexifier l'interprétation du diagramme.

Nous conseillons donc de ne pas associer une classe d'association.

XI Exercice

Question

[solution n°7 p. 39]

Dans une université, les étudiants sont inscrits à des cours.

Une inscription se rattache à une année, et donne lieu à une note sur 20 à et un certain nombre de crédits ECTS.

Les étudiants suivent au moins un cours.

Réaliser un diagramme UML en trois classes dont une classe d'association qui reprend l'ensemble des éléments présentés.

XII unicité, immuabilité, optionnalité

Objectifs

- Savoir ajouter des contraintes sur des attributs de classe ;
- Connaître les cas d'utilisation des contraintes d'unicité, d'immuabilité et d'optionnalité.

Mise en situation

Décrire une classe UML à l'aide d'attributs possédant un type est une bonne base, mais ce n'est pas toujours suffisant.

Par exemple, imaginez une classe qui décrit les modèles de smartphone : il y aura un attribut nom, un attribut marque, un attribut résolution, etc.

Est-ce-qu'il y a un sens à avoir deux smartphones qui ont le même nom ? Est-ce-que la résolution est nécessaire pour décrire un smartphone ? Est-ce-qu'un smartphone peut changer de nom ?

Dans ce module, vous allez apprendre à contraindre vos attributs UML, par exemple pour spécifier qu'un attribut ne peut pas prendre deux fois la même valeur.

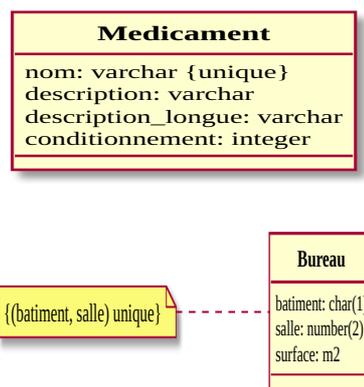
Unicité

Syntaxe

Un attribut ou un groupe d'attributs peut être annoté comme étant **unique** cela signifie que deux instances de la classe ne peuvent pas avoir la même valeur pour le ou les attributs concernés.

On ajoute le symbole `{unique}` à côté du ou des attributs concernés.

Exemple



Attention

Une classe possède en général plusieurs attributs ou groupes d'attributs uniques.

Clé

[+ Complément](#)

Si un attribut ou groupe d'attributs est unique et obligatoire, alors il permettra d'identifier de façon unique un objet de la classe (on parlera de clé en relationnel)

Une classe possède en général au moins un attribut ou un groupe d'attributs unique et obligatoire, sinon les objets ne sont pas différenciables.

Si ce groupe est composé de l'ensemble des attributs, il n'est pas nécessaire de le noter en UML.

Immuabilité

[Az Définition](#)

Une fois qu'il a été créé, un attribut immuable ne peut pas voir sa valeur changée.

On le note en UML avec `{frozen}`.

Immuabilité en UML

[👁 Exemple](#)

Personne
nom: varchar
prenom: varchar
date_naissance: date {frozen}
lieu_naissance: varchar {frozen}

Optionnalité

[Az Définition](#)

Est optionnel un attribut qui peut ne pas avoir de valeur pour certaines instances.

- Notation standard : `attribute[0..1] : type`
- Notation non standard proposée : `{optional}`

Optionnalité en UML

[👁 Exemple](#)

Autoentreprise
siret: char(14)
denomination: varchar
adresse: varchar
assureur: varchar {optional}

XIII Exercice

Question

[solution n°8 p. 39]

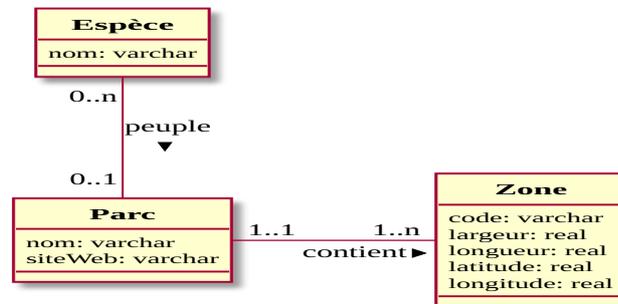
On souhaite gérer les multiples parcs de la ville de Compiègne.

Pour chaque parc on gère les espèces de plantes et d'animaux qui le peuplent. Les espèces sont identifiées de manière unique par leur nom latin qui ne change jamais.

Chaque parc dispose d'un nom qui lui est propre, et de son propre site web.

Chaque zone du parc est rectangulaire, est identifiée par un code unique et immuable, et est localisable par ses coordonnées géographiques.

On modélise la situation à l'aide du diagramme UML suivant :



À l'aide de la description de la situation, compléter le diagramme pour y inclure les informations d'unicité et d'immutabilité.

Indice :

Il est possible que des attributs soient concernés par ces deux propriétés, auquel cas on utilisera la clause `{unique frozen}`.

Indice :

Il est aussi possible que ces propriétés s'appliquent de manière jointe sur plusieurs attributs.

XIV Élaboration d'un diagramme UML

Objectifs

- Connaître les différents outils pour élaborer un diagramme UML.

Mise en situation

Si le diagramme UML est un schéma sous forme graphique, sa forme matérielle n'est absolument pas contrainte. En théorie, vous êtes libre de la choisir.

Alors comment dessiner un diagramme UML ? À la main ? Sur un logiciel de dessin ? Avec des applications spécialisées ? Faut-il avoir des compétences particulières ?

Toutes les méthodes ont leurs avantages et leurs inconvénients, mais pas besoin d'être un expert en dessin. Dans ce module, vous découvrirez les différentes pratiques pour dessiner un diagramme UML.

💡 Fondamental

L'élaboration d'un schéma UML est d'abord un outil permettant de chercher une solution, il est donc **indispensable** qu'il soit facile de le modifier, de changer d'avis, et que sa manipulation ne soit pas parasitée par de la manipulation informatique.

Papier/crayon

👉 Conseil

Pour élaborer un schéma UML la meilleure méthode est de se doter :

- d'une grande feuille (au minimum un A3 pour un petit projet de quelques classes) ;
- d'un crayon de papier et d'une gomme.

On peut aussi travailler sur un tableau (noir, blanc, numérique).

⚠ Attention

Un diagramme UML, même pour l'élaboration doit rester parfaitement visible, c'est un outil **graphique** et c'est un outil de **dialogue** s'il est mal lisible on fera des erreurs et il remplira mal son rôle.

Éditeur graphique

 Conseil

Si l'on est doté d'une bonne expérience sur un outil **graphique** alors il est possible de l'utiliser à la place du papier crayon, mais :

- il faut bien connaître l'outil pour que son attention soit portée sur la conception et non sur le fonctionnement de l'outil ;
- il faut que l'outil soit totalement souple pour permettre une expression graphique non contrainte (typiquement certains éditeurs contraignent la création afin de rester en mesure de faire de la génération automatique de code, ce qui est hors-sujet dans notre cas).

 Complément

Dia est un petit outil graphique libre et multi-plate-forme.

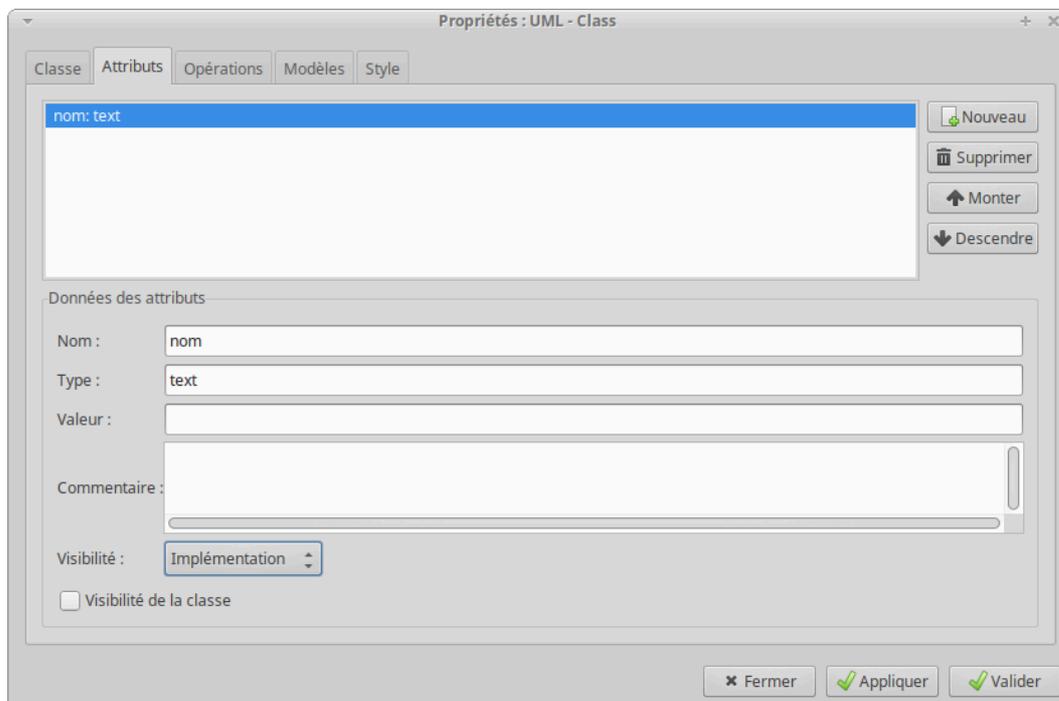
- [https://fr.wikipedia.org/wiki/Dia_\(logiciel\)](https://fr.wikipedia.org/wiki/Dia_(logiciel))
- <http://dia-installer.de/>

Il présente quelques défauts, mais il est assez facile à prendre en main et il est possible d'utiliser le mode graphique pour dessiner ce qui n'est pas prévu dans le module UML.

Quelques conseils pour bien utiliser Dia

 Complément

- Dia fixe par défaut la visibilité de tous les attributs à la valeur « *Public* », ce qui a comme conséquence de faire précéder l'attribut d'un caractère « *+*. » C'est un concept qu'on n'utilise pas en base de données relationnelle, cela surcharge donc inutilement tous les diagrammes. Il faut donc systématiquement modifier cette propriété et la mettre à la valeur « *implémentation* »).
- Utiliser la fonction Fichier > Export pour transformer votre diagramme en fichier PNG. Choisissez une résolution assez élevée si vous visez une impression papier.



Mettre la visibilité des attributs à la valeur « implémentation »

XV Exercice

L'Empereur Palpatine souhaiterait disposer d'une application pour l'aider à gérer son vaste empire galactique. Il vous a sollicité afin de mettre en œuvre une base de données de l'Empire.

Du fond de sa gorge, il vous a dit ceci :

« Dans notre magnificence, nous avons divisé la galaxie en secteurs. J'ai moi-même nommé chaque secteur. Je leur ai également attribué un code unique que nous utilisons dans les communications secrètes. Chaque secteur contient des systèmes stellaires. Comme vous le savez, un système stellaire se constitue d'une étoile qui est d'un type spectral O, B, F, G, K, M et d'un ensemble d'astres et j'aimerais d'ailleurs enregistrer les informations liées au rayon et à la masse de ces astres. »

Lexique intergalactique

- **Tellurique** : les planètes telluriques sont des planètes solides, par opposition aux planètes gazeuses.
- **Type spectral** : les étoiles présentent quatre caractéristiques principales : température de couleur, gravité à la surface, masse et luminosité. Elles permettent d'associer un type spectral à chaque étoile (Wikipédia.fr)

Question

[solution n°9 p. 40]

Réaliser le diagramme UML répondant aux besoins de l'Empereur Palpatine avec Dia.

XVI Documentation avec PlantUML

Objectifs

- Savoir décrire un diagramme UML sous forme textuelle ;
- Connaître le format PlantUML.

Mise en situation

Un diagramme de classe UML est par nature un schéma sous forme graphique, mais il y a des situations où il peut être utile de le représenter sous forme textuelle.

Imaginez : vous travaillez à plusieurs sur un diagramme UML, et chaque personne doit pouvoir modifier facilement le diagramme. Comment faire si vous utilisez des outils différents ? Si certains outils nécessitent des compétences en graphisme ?

Et surtout, comment faire pour gérer les différentes versions du diagramme ?

Une solution, c'est d'utiliser un format textuel : pratique pour comparer deux versions, et modifiable sans aucune compétence de graphisme. Ce format pourra ensuite être automatiquement transformé en image, réunissant ainsi le meilleur des deux mondes.

Dans ce module, vous apprendrez à utiliser PlantUML pour décrire vos diagrammes UML simplement avec du texte.

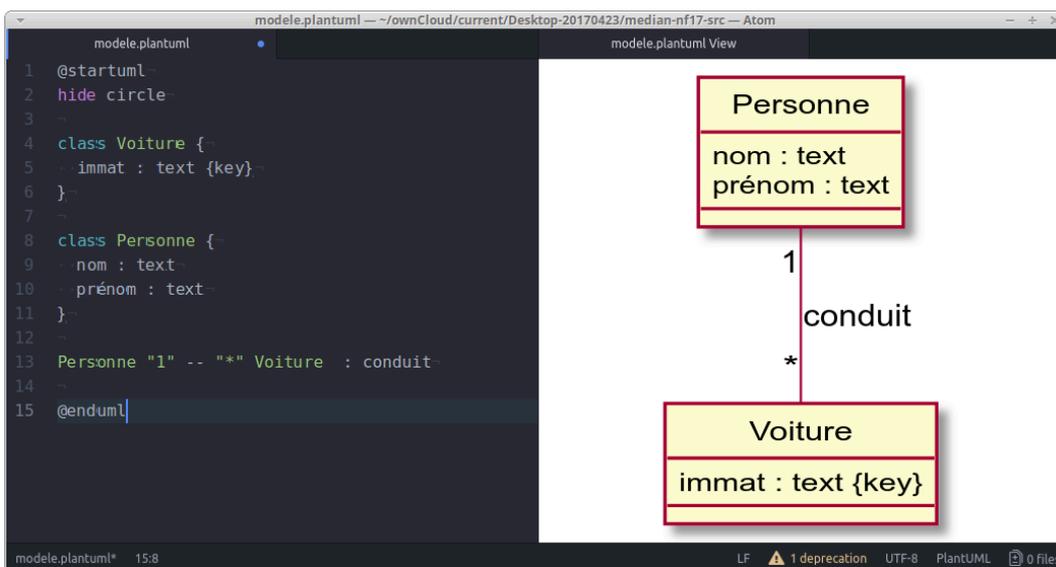
PlantUML



On conseille pour la documentation l'usage du logiciel PlantUML.

Combiné avec un dépôt Git il permet d'historiser la dynamique de conception.

PlantUML s'intègre dans de nombreux éditeurs de texte, tel que Visual Studio ou Atom.



PlantUML dans Atom

PlantUML est un outil libre et multi-plate-forme qui permet de créer des diagrammes UML à partir d'un langage textuel de description. Il est intégrable à des éditeurs comme Atom ou Visual Studio. Il peut aussi s'utiliser directement sur un serveur en ligne.

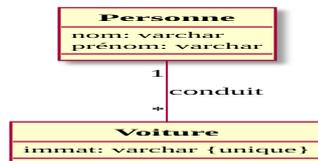
- Installation : plantuml.com/fr/running
- Documentation du diagramme de classe : plantuml.com/fr/class-diagram
- Usage direct en ligne : plantuml.com/plantuml

👁 Exemple

```

1 @startuml
2 hide circle
3
4 class Voiture {
5   immat : varchar {unique}
6 }
7
8 class Personne {
9   nom : varchar
10  prénom : varchar
11 }
12
13 Personne "1" -- "*" Voiture : conduit
14
15 @enduml

```



hide circle

🔗 Méthode

La commande `hide circle` permet de supprimer des éléments de syntaxe propre à certaines représentations objet, non utilisés en base de données.

Positionnement vertical / horizontal

🔗 Méthode

On notera la différence entre :

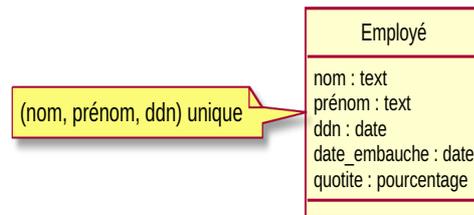
- - : à côté (un seul tiret)
- -- : en dessous (deux tirets ou plus)

L'ordre de déclaration des classes est pris en compte.

Notation des clés composées

Pour noter les clés composées de plusieurs attributs, on utilisera une note.

```
1 class Employé {  
2   nom : text  
3   prénom : text  
4   ddn : date  
5   date_embauche : date  
6   quotite : pourcentage  
7 }  
8 note left : (nom, prénom, ddn) unique
```



XVII Exercice

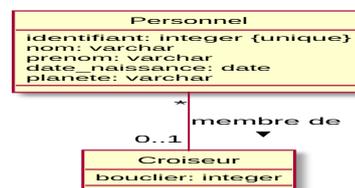
Le Capitaine Krik a pour tâche de développer une base de données l'équipage de la TarFleet.

Chaque personnel de la TarFleet a un nom, un prénom, une date de naissance, une planète d'origine et un numéro d'identifiant unique. Chaque personnel est membre d'un Croiseur qui dispose d'une puissance de bouclier

Question

[solution n°10 p. 40]

On propose le diagramme suivant pour la modélisation :



Reproduire ce diagramme de classe avec PlantUML.

XVIII Essentiel

La première étape de la conception d'une base de données est la modélisation conceptuelle.

Le formalisme UML aide à exprimer cette modélisation sous forme graphique. Comme c'est le formalisme de référence, il permet à tous les acteurs de comprendre la modélisation et d'utiliser un langage commun.

On utilise particulièrement les diagrammes de classes UML pour modéliser une situation.

D'une part, les classes permettent de décrire la structure abstraite d'un objet de la réalité, tel qu'une personne. Pour ce faire, on utilise des attributs, qui indiquent les informations que l'on veut stocker sur notre objet, par exemple le nom, la taille, la date de naissance, la profession, etc. Ces attributs ont un type et des contraintes, permettant d'affiner encore un peu plus la modélisation.

D'autre part, les classes sont toujours en interaction, et on modélise ces interactions grâce aux associations. Par exemple, le fait qu'une personne puisse être amie avec plusieurs autres personnes sera symbolisé par une association.

Enfin, si les diagrammes de classe UML sont *in fine* sous forme graphique, et qu'il existe des logiciels spécialisés pour les dessiner, il est également possible de les représenter sous forme textuelle avec des outils comme PlantUML.

XIX Quiz

Exercice 1 : Quiz - Culture

[solution n°11 p. 40]

Exercice

Que signifie UML ?

A Unified Modeling Langage

B Un Moyen Langagier

C Unique Méthode Logistique

Exercice

Quel type de diagrammes UML est utilisé pour la modélisation de bases de données ?

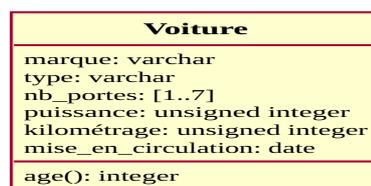
A Le diagramme de séquence

B Le diagramme de classe

C Le diagramme d'activité

Exercice

Dans le diagramme ci-dessous, qu'est-ce que age () ?



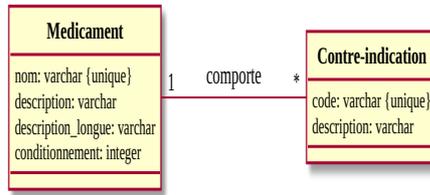
A Une classe

B Une méthode

C Un attribut

Exercice

Dans le diagramme ci-dessous, comment appelle-t-on la relation entre la classe Medicament et Contre-indication ?



- A Un lien
- B Une association
- C Une relation

Exercice 6 : Quiz - Méthode

[solution n°12 p. 42]

Exercice

Pourquoi utiliser UML pour l'étape de modélisation conceptuelle de bases de données ?

- A Parce que c'est le seul outil qui existe.
- B Parce que c'est un outil efficace de communication.
- C Parce qu'il est proche de l'implémentation finale en bases de données.

Exercice

Lorsque l'on commence à réaliser la conception en UML, quels sont les outils à privilégier ?

- A Papier A3 et crayon
- B Dia
- C PlantUML
- D Tableau et feutre

Exercice

Le travail de modélisation vise à représenter la réalité sous une forme de concepts : un modèle. Qu'est-ce qu'un bon modèle ?

- A C'est un modèle exhaustif, c'est-à-dire un modèle qui prend en compte tous les éléments de la réalité.

B

C'est un modèle qui tient compte du problème à résoudre et qui représente uniquement l'information propre à ce problème.

C

C'est un modèle avec peu de classes.

Exercice

Dans la modélisation, il est possible que des contraintes complexes apparaissent. Dans ce cas :

A

On ne les fait pas apparaître puisque l'on ne peut pas les représenter facilement : on verra cela à l'implémentation finale.

B

On les exprime explicitement dans des notes à côtés des concepts concernés.

Exercice 11 : Quiz - Modélisation

[solution n°13 p. 43]

Journal

Voici le schéma conceptuel du système d'information (très simplifié) d'un quotidien. Quelles sont les assertions vraies selon ce schéma ?

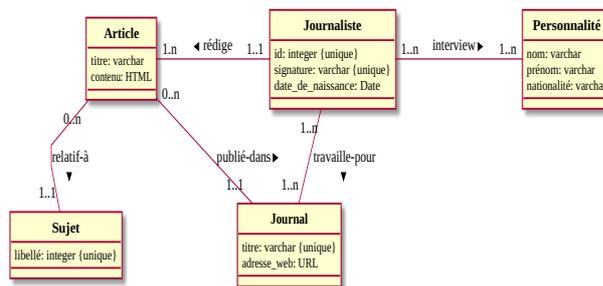


Image 11

A

Un article peut être rédigé par plusieurs journalistes.

B

Un article peut être publié plusieurs fois dans plusieurs journaux.

C

Un article peut être publié dans un journal par un journaliste qui ne travaille pas pour ce journal.

D

Il peut y avoir plusieurs articles sur le même sujet.

E

Un journaliste peut interviewer une personnalité sans faire d'article à ce propos.

Logistique

Une société de transport routier veut installer un système d'information pour rendre plus efficace sa logistique. Embauché au service informatique de cette compagnie, vous êtes donc chargé de reprendre le travail déjà effectué (c'est à dire le schéma suivant).

Quelles sont les assertions vraies selon ce schéma ?

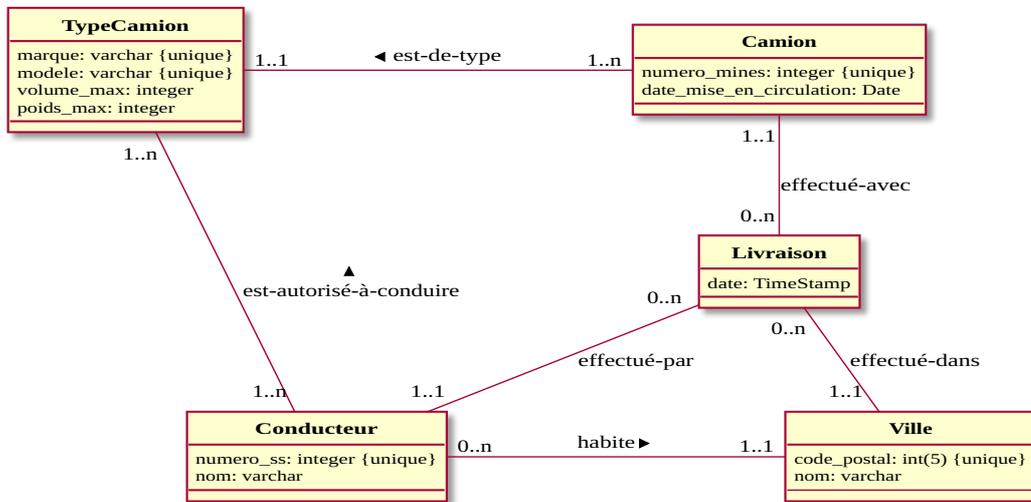


Image 12

- A** Un conducteur est autorisé à conduire plusieurs camions.
- B** Un conducteur peut conduire un camion sans y être autorisé.
- C** Il peut y avoir deux conducteurs de camion pour la même livraison.
- D** Un conducteur peut livrer sa propre ville.

Solutions des exercices

Solution n°1

[exercice p. 6]

Soit le modèle conceptuel suivant représentant des visites dans un centre médical. Quelles sont les assertions vraies selon ce schéma ?

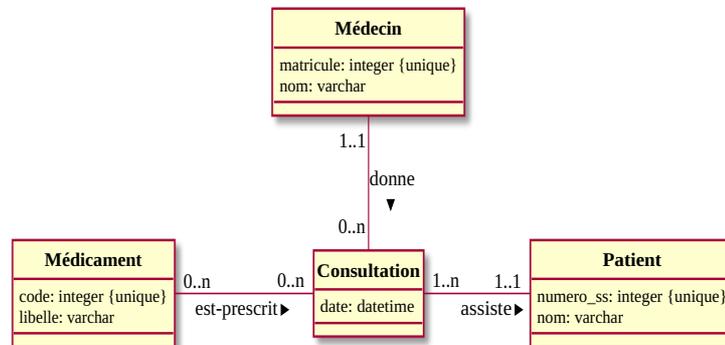


Image 13

A Un patient peut effectuer plusieurs visites.

B Tous les patients ont effectué au moins une consultation.

C Un médecin peut recevoir plusieurs patients pendant la même consultation.

D Un médecin peut prescrire plusieurs médicaments lors d'une même consultation.

E Deux médecins différents peuvent prescrire le même médicament. Oui, mais pas au sein de la même consultation, puisqu'une consultation n'est effectuée que par un seul médecin.

Solution n°2

[exercice p. 9]



Solution n°3

[exercice p. 13]



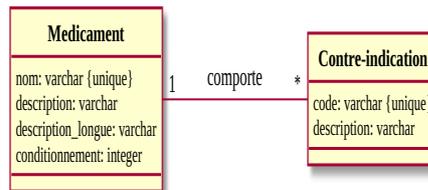
Solution n°4

[exercice p. 13]



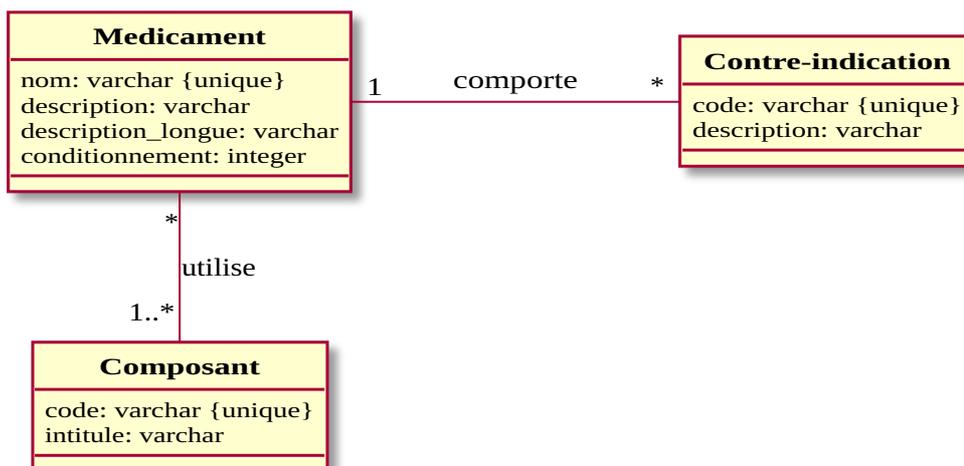
Solution n°5

[exercice p. 17]

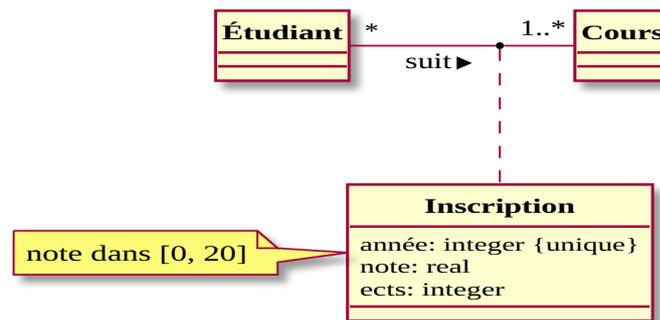


Solution n°6

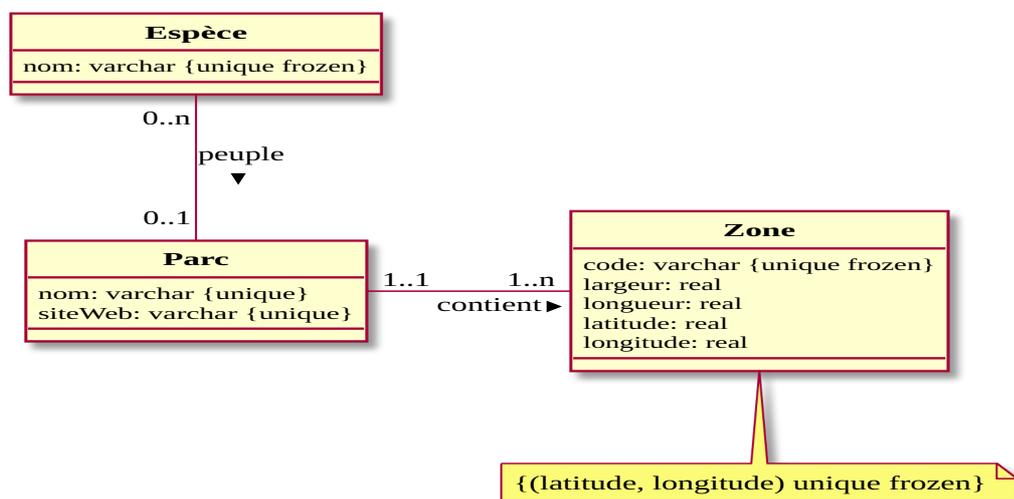
[exercice p. 17]



Solution n°7

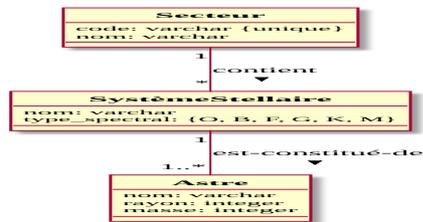


Solution n°8



Solution n°9

[exercice p. 27]



Solution n°10

[exercice p. 31]

On peut obtenir ce diagramme avec le script PlantUML suivant.

```

1 @startuml
2 hide circle
3 class Personnel {
4   identifiant: integer {unique}
5   nom: varchar
6   prenom: varchar
7   date_naissance: date
8   planete: varchar
9 }
10 class Croiseur {
11   bouclier: integer
12 }
13 Personnel "*" -- "0..1" Croiseur : membre de >
14 @enduml
15
  
```

Solution n°11

[exercice p. 33]

Exercice

Que signifie UML ?

A Unified Modeling Langage

B Un Moyen Langagier

C Unique Méthode Logistique

Exercice

Quel type de diagrammes UML est utilisé pour la modélisation de bases de données ?

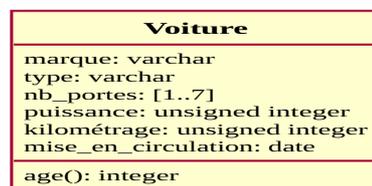
A Le diagramme de séquence

B Le diagramme de classe

C Le diagramme d'activité

Exercice

Dans le diagramme ci-dessous, qu'est-ce que `age()` ?



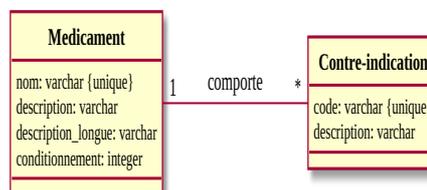
A Une classe

B Une méthode

C Un attribut

Exercice

Dans le diagramme ci-dessous, comment appelle-t-on la relation entre la classe **Medicament** et **Contre-indication** ?



A Un lien

B Une association

C Une relation

Solution n°12

Exercice

Pourquoi utiliser UML pour l'étape de modélisation conceptuelle de bases de données ?

- A** Parce que c'est le seul outil qui existe.
- B** Parce que c'est un outil efficace de communication.
- C** Parce qu'il est proche de l'implémentation finale en bases de données.

🔍 UML est un formalisme de modélisation et un outil efficace de communication flexible. Ce n'est pas le seul outil de formalisation : on peut utiliser aussi l'Entité-Association pour réaliser la modélisation mais celui-ci tend à ne plus être utilisé. UML est indépendant de tout système d'implémentation, ce qui est important à l'étape de conception.

Exercice

Lorsque l'on commence à réaliser la conception en UML, quels sont les outils à privilégier ?

- A** Papier A3 et crayon
- B** Dia
- C** PlantUML
- D** Tableau et feutre

🔍 On préférera les outils de communication qui laissent le plus de liberté d'expression et qui ne contraignent pas l'usage : le papier et le crayon (ou le tableau et la craie) sont mieux adaptés pour cela. À ce stade, on préfère faire évoluer la conception rapidement plutôt que de faire des diagrammes parfaitement avec des outils.

Exercice

Le travail de modélisation vise à représenter la réalité sous une forme de concepts : un modèle. Qu'est-ce qu'un bon modèle ?

- A** C'est un modèle exhaustif, c'est-à-dire un modèle qui prend en compte tous les éléments de la réalité.

B

C'est un modèle qui tient compte du problème à résoudre et qui représente uniquement l'information propre à ce problème.

C C'est un modèle avec peu de classes.

Exercice

Dans la modélisation, il est possible que des contraintes complexes apparaissent. Dans ce cas :

A

On ne les fait pas apparaître puisque l'on ne peut pas les représenter facilement : on verra cela à l'implémentation finale.

B

On les exprime explicitement dans des notes à côtés des concepts concernés.



On doit toujours faire apparaître dans la modélisation toutes les informations nécessaires de la situation: l'étape de modélisation est la plus importante et les détails manquant en amont causent des problèmes techniques en aval.

Dans le cas où les éléments d'UML ne permettent pas d'ajouter de formaliser la contrainte, on utilise les notes.

Solution n°13

[exercice p. 35]

Journal

Voici le schéma conceptuel du système d'information (très simplifié) d'un quotidien. Quelles sont les assertions vraies selon ce schéma ?

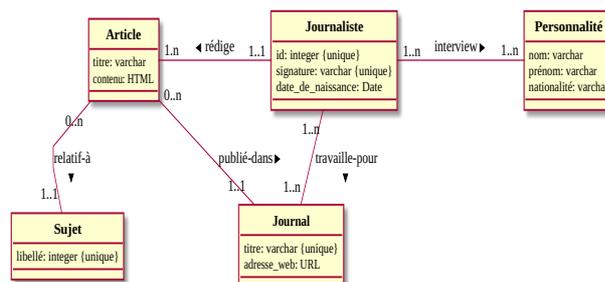


Image 14

A Un article peut être rédigé par plusieurs journalistes.

B

Un article peut être publié Dans ce modèle il n'y a aucune information sur le journal plusieurs fois dans plusieurs qui publie les articles, donc aucune contrainte à ce propos.

C

Un article peut être publié dans un journal par un journaliste qui ne travaille pas pour ce journal.

On sait que chaque article est rédigé par un journaliste et est publié dans un journal.

Mais rien ne dit que le journaliste qui a rédigé l'article travaille dans ce journal ci.

D

Il peut y avoir plusieurs articles sur le même sujet.

E

Un journaliste peut interviewer une personnalité sans faire d'article à ce propos.

Tout journaliste a interviewé au moins une personnalité et écrit au moins un article, mais rien ne dit que c'est à ce propos.

Logistique

Une société de transport routier veut installer un système d'information pour rendre plus efficace sa logistique. Embauché au service informatique de cette compagnie, vous êtes donc chargé de reprendre le travail déjà effectué (c'est à dire le schéma suivant).

Quelles sont les assertions vraies selon ce schéma ?

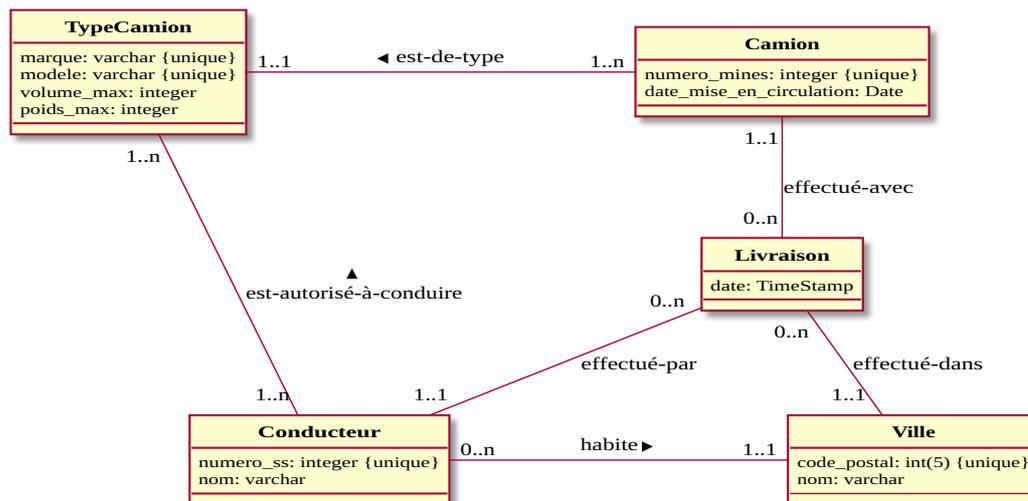


Image 15

A

Un conducteur est autorisé à conduire plusieurs camions.

B

Un conducteur peut conduire un camion sans y être conducteur et les types de camion qu'il est autorisé à conduire.

Il n'y a pas de contrainte entre les livraisons qu'effectue un camion et les types de camion qu'il est autorisé à conduire.

C

Il peut y avoir deux conducteurs de chaque livraison n'est associée qu'à un seul camion pour la même livraison.

de chaque livraison n'est associée qu'à un seul camion et un seul conducteur.

D

Un conducteur peut livrer sa propre ville. Rien n'indique le contraire.

Abréviations

OMG : Object Management Group

UML : Unified Modeling Language

Index

Association	14, 18
Attribut.....	10, 18
Classe.....	7
Conceptuel	4
OMG.....	4
Propriété.....	10, 18
UML	4, 7, 10, 14, 18

