Exercices UML
Programmation orientée Objet

TABLE DES MATIERES

Section 1 Un distributeur de billets

1.1 But

- Diagramme use case

1.2 Enoncé

On considère un distributeur automatique de billet (DAB) avec les fonctions simplifiées suivantes :

- Pour tout client de la banque il autorise :
  - La consultation d'argent du compte
  - Le dépôt d'argent (chèque ou liquide dans une enveloppe)
- Le distributeur délivre de l'argent pour tout porteur de carte (Visa ou carte bancaire) même s'il n'est pas client
- Toute transaction est protégée et nécessite une authentification
- Un opérateur de maintenance :
  - Récupère toute carte avalée par le distributeur
  - Recharge en billets le distributeur
  - Récupère les dépôts d'argent

Modéliser cette situation par un diagramme de cas d'utilisation.

Il est possible que vous commenciez par un autre dessin, non UML, pour identifier les acteurs clés, les actes clés …

Section 2 Gestion d'articles dans un stock

2.1 But

- Diagramme use case

2.2 Enoncé

Un commerçant dispose d'un système de gestion de stock d'articles, avec les fonctionnalités suivantes :

- Edition de la fiche d'un fournisseur
- Ajout d'un nouvel article délivré par un fournisseur. La fiche fournisseur est éditée si existe déjà, sinon on peut créer le fournisseur
- Gestion de l'inventaire. Depuis cet écran on peut soit imprimer l'inventaire, effacer un article, éditer la fiche d'un article.

Modéliser cette situation par un diagramme de cas d'utilisation.

Section 3 Achat dans un magasin

3.1 But

- Diagramme use case

3.2 Enoncé

Dans un magasin, le processus de vente est le suivant :

Le client entre, voyage dans les rayons, demande éventuellement des renseignements, ou procède à des essais, prend des articles si le stock est suffisant, passe à la caisse et effectue le règlement (tout moyen accepté (monnaie, chèque, carte). Il peut éventuellement bénéficier d'une réduction.

Réaliser un diagramme de cas d'utilisation de cette situation.

Section 4 Gestion de salle de cours

4.1 But

- Diagramme use case

4.2 Enoncé

Dans un établissement scolaire, on désire gérer la réservation des salles de cours et de matériel pédagogique (ordi, vidéo projecteur). Seuls les enseignants sont habilités à effectuer des réservations (sous réserve de disponibilité de salle ou matériel).

Par ailleurs le planning des salles peut être consulté par tout le monde (enseignants et étudiants)

Par contre le récapitulatif horaire par enseignant (calculé à partir du planning des salles) ne peut être consulté que par les enseignants.

Enfin il existe pour chaque formation un enseignant responsable qui seul peut consulter le récapitulatif horaire pour l'ensemble de la formation.

Réaliser de diagramme de cas d'utilisation.

Section 5 Travail du réceptionniste dans un hôpital

5.1 But

- Diagramme use case

5.2 Enoncé

Son travail est assez varié :

- Il réalise l'admission des patients dont leur venue a été planifiée par avance, ceux qui viennent pour une seule journée, ceux qui auront besoin d'un lit pour plusieurs jours
- Pour ceux qui ont besoin d'un lit le réceptionniste leur en affecte un
- Il réalise la planification de l'admission des patients
- Il réalise la planification des rendez-vous des patients (pour peut être convenir d'une future admission)
- Il complète le dossier des patients

Réaliser de diagramme de cas d'utilisation.

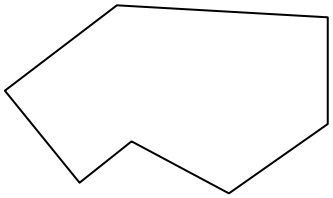
Section 6 Le polygone

6.1 But

- Un 1er diagramme de classe simple

6.2 Enoncé

Représenter le diagramme de classe modélisant un polygone en tenant compte de la relation entre les points du polygone



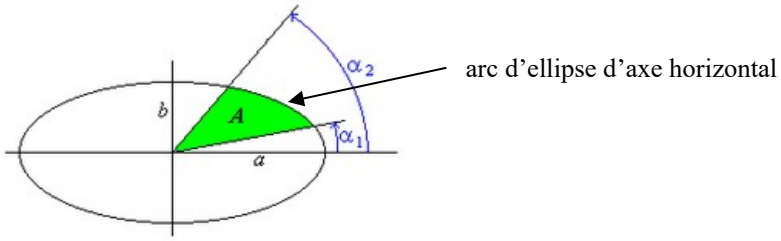
Section 7 Les figures

7.1 But

- Création d'un diagramme de classe avec de l'héritage

7.2 Enoncé

Proposer un diagramme de classe représentant différentes figures géométriques : point, ligne, arc de cercle, rectangle, carré, ellipse d'axe horizontal, cercle, arc d'ellipse d'axe horizontal



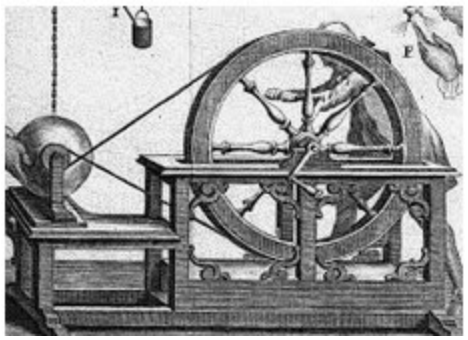
Section 8 Machine électrique

8.1 But

- Diagramme de classe

8.2 Enoncé

Représenter le diagramme de classe d'une machine électrique. Les machines électriques peuvent se classer en machines à courant alternatif et en courant continu, voire les 2. Une machine en courant alternatif peut être synchrone ou inductive.



Section 9 Système de réservation

9.1 But

- Diagramme plus complexe
- Travailler en groupe

9.2 Enoncé

Construire un diagramme de classe représentant le système de réservations suivant :

client, réservation, passager, vol, compagnie, aéroport, escale, ville.

Ajoutez des attributs et des méthodes à chaque classe.

Section 10 Analyse de code

10.1 But

- Comprendre un code
- Prédire le comportement

10.2 Enoncé

On considère le code java suivant:

```
public class F {
	public void f() {
		System.out.print("Passage dans F.f() ");
		this.g();
	}

	protected void g() {
		System.out.print("Passage dans F.g() ");
	}
}

public class Fbis extends F {
	public void f() {
		System.out.print("Passage dans Fbis.f() ");
		this.g();
	}

	protected void g() {
		System.out.print("Passage dans Fbis.g() ");
		super.f();
	}
}
```

Dans le main :

```
F fObj = new Fbis();
fObj.f();
```

Trouver la bonne réponse:

(a) il ne peut pas être compilé

(b) il boucle à l'exécution

(c) sa sortie sera: Passage dans F.f() Passage dans F.g()

(d) sa sortie sera: Passage dans Fbis.f() Passage dans Fbis.g() Passage dans F.f() Passage dans F.g()

Section 11 Le Flipper

11.1 But

- Diagramme complet

11.2 Enoncé

Un flipper est composé d'un jeu de 3 billes, d'un panneau de score, et d'obstacles.

5 catégories d'obstacle peuvent se trouver dans le flipper :

- la paroi,
- le trou qui fait disparaître la bille,
- le ressort qui la fait rebondir,
- le champignon de valeur incrémente le score de la valeur indiquée sur l'obstacle
- et le champignon à ressort qui se comporte comme un champignon et un ressort.

Réaliser le diagramme de classe UML.

Veiller à bien représenter le fait qu'un champignon à ressort est un champignon …. à ressort.

Section 12 Programmation objet

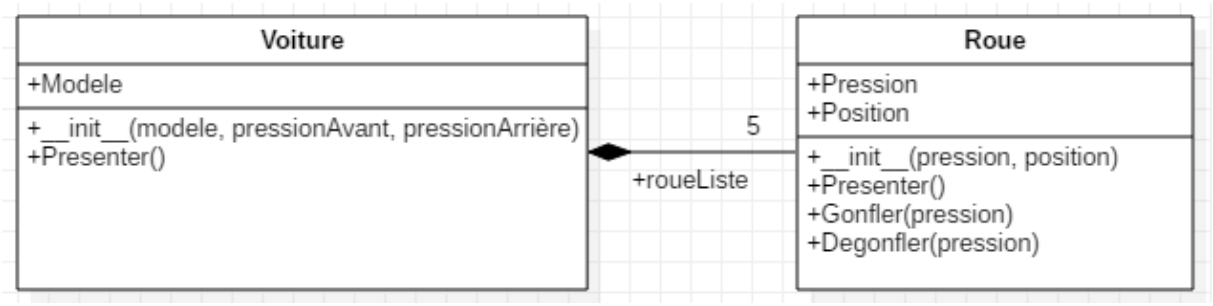
12.1 But

- Ecriture de classes associées

12.2 Enoncé

Les premiers pas en conception Objet

Le diagramme UML de classe suivant représente la conception et relation entre une voiture et ses roues



Une Voiture est caractérisée par son modèle (ex : C3, 207, SMART …).

Elle possède 5 roues (roue de secours comptée).

Les roues avant sont gonflées à la pressionAvant, idem pour l'arrière.

La roue de secours est gonflée avec la plus forte des 2 pressions.

La méthode Presenter() réalise simplement un print de son modèle et de ses 5 roues.

Une roue est caractérisée par sa pression en bars et sa position : avant gauche , avant droit, arrière gauche, arrière droit, roue de secours.

La méthode Presenter() réalise simplement un print de sa pression et de sa position

Ecrire le code correspondant en PHP ou Java ou C# :

- Pour décrire la classe Voiture
- Pour décrire la classe Roue

Puis créer un objet Voiture C3 , pression avant à 2.5 et pression arrière à 2.2

Afficher les caractéristiques de cette voiture pour obtenir quelque chose comme ceci :

Voiture C3
Pneu Avant Gauche pression 2.5
Pneu Avant Droit pression 2.5
Pneu Arrière Gauche pression 2.2
Pneu Arrière Droit pression 2.2
Pneu De secours pression 2.5

Pour démarrer il est possible d'utiliser le squelette de code suivant (écrit en python !) :

```
#!/usr/bin/python
#-*- coding: utf-8 -*-

class Voiture:

    def __init__(self, modele, pressionAvant, pressionArrière):
        # to do

    def Presenter(self, ):
        # to do

class Roue:
        self.Pression = None
        self.Position = None

    def __init__(self, pression, position):
        # to do

    def Presenter(self, ):
        # to do

    def Gonfler(self, pression):
        # to do

    def Degonfler(self, pression):
        # to do
```

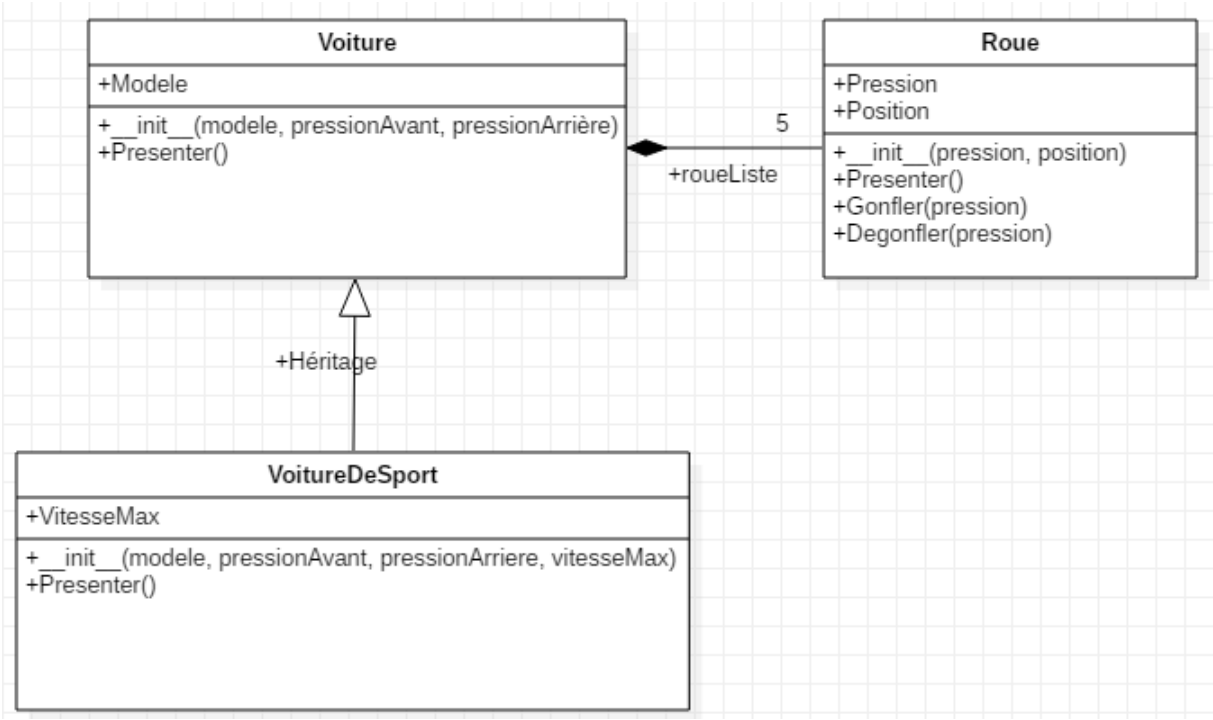
Section 13 Héritage

13.1 But

- Implémenter une classe enfant dans l'exercice précédent

13.2 Enoncé

Notre conception fait apparaître les Voitures de sport comme étant un cas particulier de Voiture. Le diagramme de classes UML en fait état :



Dans le code de l'exercice précédent, coder la classe dérivée VoitureDeSport.

Le constructeur __init__() de Voiture de sport doit appeler le constructeur de Voiture par

Voiture.__init__()

Presenter() de VoitureDeSport doit afficher en plus la vitesse max.

Section 14 Le polymorphisme

14.1 But

- Comprendre le polymorphisme d'objets issus de la même classe de base

14.2 Enoncé

Continuer le code de l'exercice précédent "Héritage".

En dehors des codes de classe, créer une liste qui contient 2 objets Voiture et 2 objets VoitureDeSport.

Ecrire ensuite une boucle for sur cette liste pour Presenter chaque objet

Section 15 Un avion en détresse

15.1 But

- Dessiner un diagramme d’objets

15.2 Enoncé

Un objet nommé b747 de classe Avion et en état _ détresse _est en relation avec luna, une tour de contrôle.

Un ensemble d'autres avions anonymes dont l'état est _ à terre _sont aussi liés à luna.

La tour de contrôle communique avec p123, une caserne de pompiers.

Dessinez le diagramme d'objets correspondant à la situation décrite ci-dessus.

Section 16 Instanciation d'un diagramme de classes

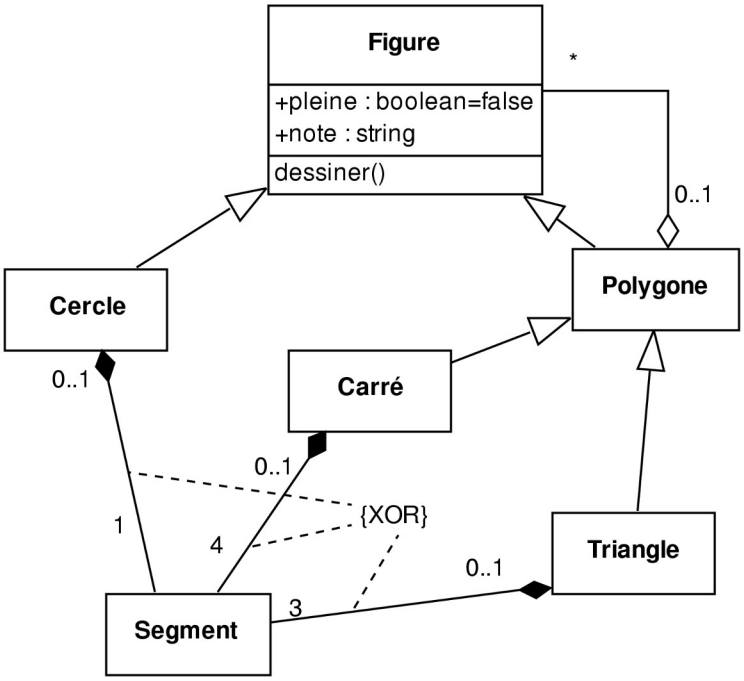
16.1 But

- Diagramme d'objets à partir d'un diagramme de classe
- Vérifier une conception

16.2 Enoncé

Considérez le diagramme de classes ci-dessous.

Le _ XOR _est une contrainte indiquant que l'on peut avoir un segment lié à une figure, mais pas à plusieurs en même temps.

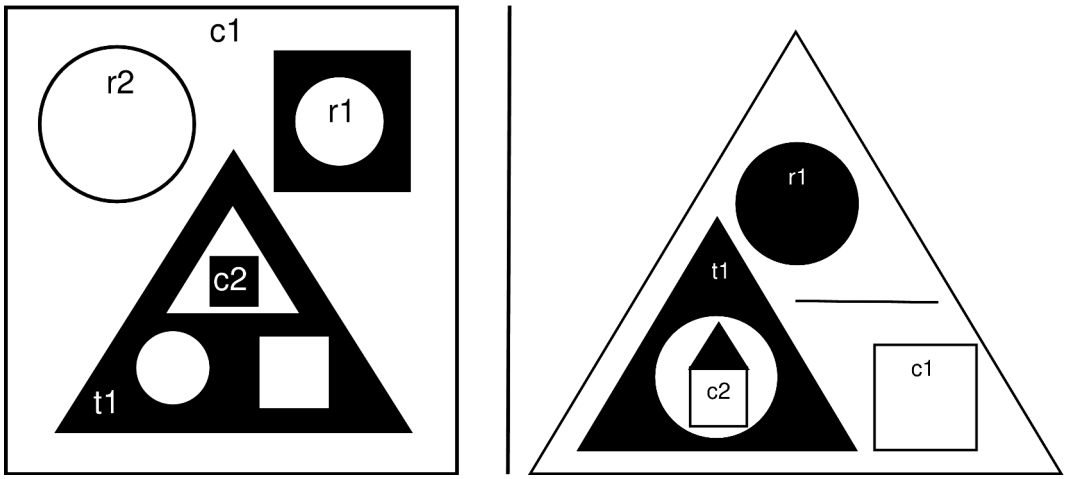


Les schémas ci-dessous illustrent des cas particuliers d'imbrication de certaines figures dans d'autres.

Si une figure est représentée directement à l'intérieur d'une autre, c'est qu'elle est nécessairement imbriquée dans cette dernière.

Le nom des figures est parfois indiqué directement à l'intérieur.

Les figures en noir sont considérées comme pleines, les autres non.



Question : Pour chacun des deux schémas ci-dessus, indiquez si l'agencement des objets représentés est conforme au diagramme des classes ci-dessus.

Question : Quand c'est le cas, représentez le diagramme des objets correspondant à la figure, mais sans représenter les segments.