**UML**

**Les diagrammes de classes**

# But

Le but de ces exercices est d’entraîner la modélisation d’une architecture de classes et de représenter cette architecture sous forme de diagrammes de classes en UML.

# Prérequis

Connaître la représentation d’une classe en UML, ainsi que les trois types de liens : l’association, l’agrégation et la composition.

# Diagrammes de classes

## Le pays

Modéliser la situation suivante : Un pays possède une capitale.

À noter que dans cet exercice, il ne faut pas confondre le statut de capitale (ce que l’on cherche, et qui est lié au pays) et celui de ville : une ville ne disparaît pas lorsque le pays est dissolu).

D’autre part, la modélisation proposée est volontairement simple et ne tient pas compte de particularités : une capitale partagée par deux pays (Nicosie, ev. Jérusalem), ou un pays contenant plusieurs capitales (Afrique du Sud et 10 autres pays).

C’est d’ailleurs souvent les cas particuliers qui compliquent singulièrement la modélisation et pour lesquels il faut faire des choix.



## Repas de famille

Permets d’introduire la notion de *rôle*.

Modéliser la situation suivante : Les membres d’une famille mangent avec une fourchette.

Une personne peut-elle faire partie de plusieurs familles ? On pourrait décider que oui.



## Système de fichier

Modéliser la situation suivante : Un répertoire d’un système de fichiers peut contenir des fichiers et/ou des répertoires.



Voilà un exemple de composition mais dont le propriétaire pourrait changer en cours d’existence : un fichier peut appartenir à un répertoire, puis à un autre.

## Physique moléculaire

Modéliser la situation suivante : Une molécule est constituée d’au moins deux atomes. Si un des atomes se détache, la molécule disparaît.



Voilà un exemple d’agrégation : si un atome disparaît, la molécule disparaît, mais la durée de vie des atomes n’est pas liée à une molécule.

## La voiture

Modéliser la situation suivante : Une voiture possède 4 roues et éventuellement une roue de secours.



Ou



## Le document

Nécessite d’avoir des notions d’héritage.

Modéliser la situation suivante : Un document est constitué d’éléments textuels, graphiques ou de groupes d’éléments.



## La classe

Modéliser la situation suivante : Une salle de classe contient jusqu’à 30 étudiants et un enseignant. Parfois, s’ajoutent à la classe une ou deux personnes en tant qu’auditeurs.



## Lego

Une construction Lego est constituée de briques de différentes formes. Chaque forme est d’une couleur parmi les possibilités suivantes : rouge, bleu, noir, jaune et blanc.

Chaque brique a une apparence particulière. Pour simplifier, les briques suivantes existent :

* Brique de 4x2
* Brique de 2x2
* Brique de 1x1
* Plaque de 4x2



## Pokémon Go

Modéliser la situation suivante : Un joueur de Pokémon Go, lorsqu’il atteint le niveau 5, fait partie d’une équipe parmi les suivantes :

* équipe jaune (intuition)
* équipe bleue (sagesse)
* équipe rouge (bravoure)

Une équipe peut posséder des arènes.

Le joueur peut posséder des pokéballs et des Pokémons de différentes espèces.



On peut discuter de l’agrégation entre Pokémon et Espèce de Pokémon, mais le lien me semble suffisamment fort pour le justifier.

## Excursions touristiques

Modéliser avec le diagramme de classes UML la situation suivante : Un tour opérateur possède cinq bus pour faire des excursions touristiques. Chaque bus, conduit par un chauffeur, accueil un maximum de 25 passagers. À chaque bus est assigné un des itinéraires proposés par le tour opérateur. Pour que l’excursion ait lieu, il faut un minimum de 5 passagers.



## Le jeu de cartes

Modéliser l’architecture d’une application de jass (chibe), qui se joue avec quatre joueurs et un jeu classique de 36 cartes (cœur, trèfle, carreau, pique).

Les notions de match, de parties, de score et de stratégie n’ont pas à être modélisées.



# Diagrammes de séquences

## Le bras robotique

Un robot, constitué d’un bras robotique articulé muni à son extrémité d’une pince est programmé pour aller chercher des pièces.

Le bras articulé peut se déplier et se replier.

La pince peut s’ouvrir et se fermer.

Lorsqu’un opérateur actionne le robot pour aller chercher une pièce, le bras se déplie, la pince se ferme pour attraper la pièce, puis le bras se replie et la pince s’ouvre pour délivrer la pièce.

Représentez cette action au moyen d’un diagramme de séquence.



## La librairie en ligne

Le client d’une librairie en ligne est en train de constituer son panier.

Lorsqu’il ajoute un livre dans son panier, le panier demande au catalogue de lui fournir le prix de ce livre. Pour cela, le catalogue doit chercher le livre en question dans ses données, puis, le livre trouvé, demander au livre son prix. Cette information est retournée au panier, qui met à jour son montant total.

Représentez cette action au moyen d’un diagramme de séquence.



## Réservation d’une place au concert

Le client d’un système de réservation de places en ligne souhaite réserver sa place pour un concert à une date donnée.

Il communique au site de réservation la date qui l’intéresse. Le site prend contact avec le serveur de billets, qui vérifie (et retourne) quelles sont les places disponibles à la date donnée. Le site de réservation propose ces places au client.

Le client choisi une place et le site de réservation contacte le serveur de billets pour réserver cette place. Celui-ci contacte alors le service de carte de crédit, afin de facturer le montant sur le compte du client.

Deux cas peuvent alors se produire :

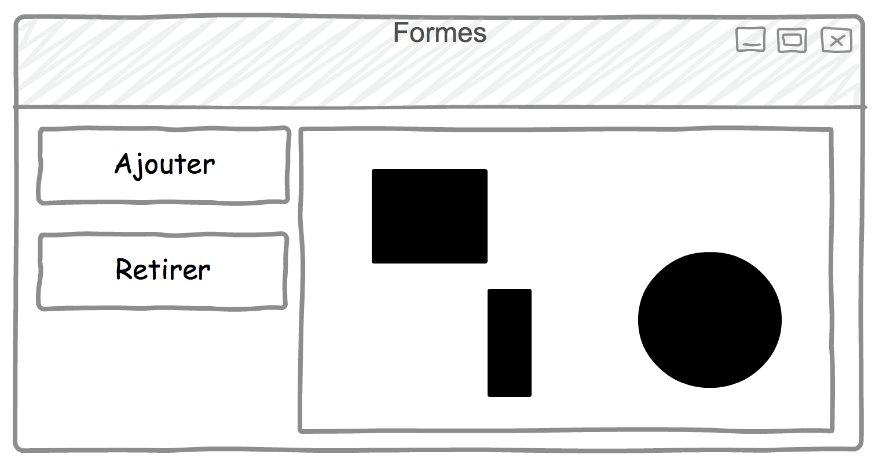
* le montant est correctement facturé : auquel cas le serveur de billets procède à la réservation et le site de réservation envoie une confirmation au client.
* Le paiement est refusé par le service de carte de crédit : le serveur de billets ne réserve aucune place et le site de réservation indique au client que la réservation est annulée.

Représentez cette action au moyen d’un diagramme de séquence.



## Formes graphiques

Une future application graphique sera constituée de deux boutons et d’une zone de dessin.



Lorsque l’utilisateur clique sur le bouton Ajouter, une nouvelle forme aléatoire est affichée dans la zone de dessin.

Lorsque l’utilisateur clique sur le bouton Retirer, la dernière forme affichée est effacée de la zone de dessin et supprimée.

Les objets graphiques suivants seront créés au démarrage de l’application :

btnAjouter : Button, btnRetirer : Button, zoneDessin : Canvas

Enfin, les formes sont de type Forme, crées par un appel à la méthode créerFormeAléatoire, de la classe FormeFactory.

Représentez les actions d’ajouter et de retirer une forme dans un diagramme de séquence.

