

TD 1 (DCU)

Questions de cours (Identification des acteurs et de cas d'utilisation simples)

Considérons une station-service de distribution d'essence. Les clients se servent de l'essence et le pompiste remplit les cuves.

Question : Le client se sert de l'essence de la façon suivante : il prend un pistolet accroché à une pompe et appuie sur la gâchette pour prendre de l'essence. Qui est l'acteur du système ? Est-ce le client, le pistolet ou la gâchette ?

Question : Jojo, dont le métier est pompiste, peut se servir de l'essence pour sa voiture. Pour modéliser cette activité de Jojo, doit-on définir un nouvel acteur ? Comment modélise-t-on ça ?

Question : Lorsque Jojo vient avec son camion citerne pour remplir les réservoirs des pompes, est-il considéré comme un nouvel acteur ? Comment modélise-t-on cela ?

Question : Certains pompistes sont aussi qualifiés pour opérer des opérations de maintenance en plus des opérations habituelles des pompistes telles que le remplissage des réservoirs. Ils sont donc réparateurs en plus d'être pompistes. Comment modéliser cela ?

Exercices sur les relations entre CUs

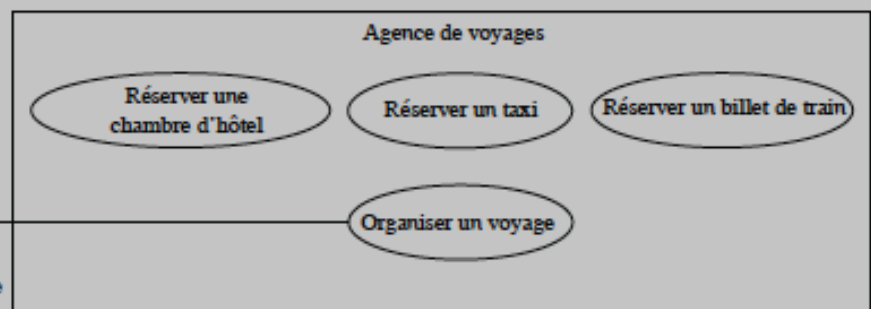
A. Agence de voyage

Choisissez et dessinez les relations entre les cas suivants :

1. Une agence de voyages organise des voyages où l'hébergement se fait en hôtel. Le client doit disposer d'un taxi quand il arrive à la gare pour se rendre à l'hôtel.

Figure 1.18
Diagramme
incomplet des cas
d'utilisation
d'une agence
de voyages.


Agent de
voyages



2. Certains clients demandent à l'agent de voyages d'établir une facture détaillée. Cela donne lieu à un nouveau cas d'utilisation appelé « Établir une facture détaillée ». Comment mettre ce cas en relation avec les cas existants ?
3. Le voyage se fait soit par avion, soit par train. Comment modéliser cela ?

B. Suivi de commande

Soient les cas d'utilisation suivants :

- Passer une commande
- Passer une commande urgente
- Suivre une commande
- Valider l'utilisateur
- Passer une commande
- Expédier commande totale ou partielle

Le suivi de la commande désigne le processus complet, du passage à l'expédition. Il peut toutefois arriver qu'une commande passée ne soit pas envoyée. Passer une commande urgente est un cas particulier de passer une commande. Pour passer une commande, il faut nécessairement valider l'utilisateur.

Question : Donner le diagramme de cas d'utilisation sans représenter les acteurs

Exercice sur la spécification des besoins avec un DCU

Préparation à l'achat d'un cheval

Un système informatique doit permettre à des acheteurs potentiels de préparer l'achat de chevaux (mais pas l'achat proprement dit). L'achat d'un cheval concerne soit une jument soit un étalon. Dans le premier cas, on doit impérativement examiner l'état de maternité du cheval, et éventuellement vérifier que la jument n'a pas un jeune poulain en ce moment. Que l'on souhaite acheter un étalon ou une jument, on doit effectuer un examen des vaccinations. En outre l'acheteur peut souhaiter, lors de la préparation de son achat, consulter le caractère du cheval ou bien en connaître la robe. Toutes les informations en rapport à la filiation d'un cheval sont obtenues en consultant la base de données externe des haras nationaux.

Question : Donnez un diagramme de cas d'utilisation pour le système de préparation avant achat.

Exercices sur la description textuelle

- A. Faire une lecture méthodique (observer, analyser et interpréter) de la description textuelle du cas d'utilisation « S'inscrire sur un site pour devenir client d'une galerie d'art » suivant :

Scénario Nominal :

- 1 L'internaute saisit son nom, son prénom, son adresse email;
- 2 Le système valide ces informations (bien construites);
- 3 Le système enregistre le nouveau client;
- 4 Le système signale au client que tout s'est bien passé. (Important : le retour vers l'acteur est quasi obligatoire dans tout scénario, sous peine de définir un système qui n'est pas assez explicite.

Scénario Alternatif :

-A1 : Données non valides

L'enchaînement A1 démarre au point 2 du scénario nominal;

21. Le système indique à l'internaute que les données sont invalides.

22. Le scénario nominal reprend au point 1.

A2 : Client déjà enregistré

L'enchaînement A2 démarre au point 3 du scénario nominal;

31. Le système indique à l'internaute qu'un client avec les mêmes informations est déjà connu du système.

32. Le système propose à l'internaute de ressaisir les informations.

33. Le scénario nominal reprend au point 1.

On vous envoie la description textuelle suivante. Donner les critiques et apporter les corrections nécessaires.

Après cette lecture, examiner et comparer les variantes suivantes :

Version 1 :

1. Le client saisit son numéro de commande. Le système détecte que ce numéro correspond au numéro gagnant, inscrit l'utilisateur et le numéro de commande comme gagnant du mois, envoie un courrier électronique au responsable des ventes, félicite le client et lui donne les instructions nécessaires pour retirer son prix.

Version 2 :

1. Le client saisit son numéro de commande.
2. Le système détecte que ce numéro correspond au numéro gagnant, inscrit l'utilisateur et le numéro de commande comme gagnant du mois, envoie un courrier électronique au responsable des ventes, félicite le client et lui donne les instructions nécessaires pour retirer son prix.

Version 3 :

1. Le client saisit son numéro de commande.
2. Le système détecte que ce numéro correspond au numéro gagnant,
3. Le système inscrit l'utilisateur et le numéro de commande comme gagnant du mois, envoie un courrier électronique au responsable des ventes, félicite le client et lui donne les instructions nécessaires pour retirer son prix.

Version 4 :

1. Le client saisit son numéro de commande.
2. Le système détecte que ce numéro correspond au numéro gagnant,
3. Le système inscrit l'utilisateur et le numéro de commande comme gagnant du mois et envoie un courrier électronique au responsable des ventes,
4. Le système félicite le client et lui donne les instructions nécessaires pour retirer son prix.

Version 5 :

1. Le client saisit son numéro de commande.
2. Le système détecte que ce numéro correspond au numéro gagnant,
3. Le système inscrit l'utilisateur et le numéro de commande comme gagnant du mois,
4. Le système envoie un courrier électronique au responsable des ventes,
5. Le système félicite le client et lui donne les instructions nécessaires pour retirer son prix.

B. Corriger la description textuelle d'un CU envoyé par un camarade

Cas d'utilisation 25 : ouvrir une session

Ce cas d'utilisation décrit le processus par lequel les utilisateurs se connectent au système de traitement des commandes. Il vise aussi à établir les autorisations d'accès aux différentes catégories d'utilisateurs (clients, employés, ...).

Scénario :

1. *Le cas d'utilisation débute lorsque l'utilisateur démarre l'application*
2. *Le système affiche l'écran d'ouverture de session*
3. *L'utilisateur saisit un nom d'utilisateur et un mot de passe*
4. *Le système vérifie les informations*
5. *Le système définit les autorisations d'accès*
6. *Le système affiche l'écran principal*
7. *L'utilisateur sélectionne une fonction*
8. *Tant que l'utilisateur ne sélectionne pas la sortie, boucler :*
9. *Si l'utilisateur sélectionne Passer une commande, utiliser Passer une commande*
10. *Si l'utilisateur sélectionne Retourner le produit, utiliser Retourner le produit*
11. *Si l'utilisateur sélectionne Annuler la commande, utiliser Annuler la commande*
12. *Si l'utilisateur sélectionne Obtenir l'état de la commande, utiliser Obtenir l'état de la commande*
13. *Si l'utilisateur sélectionne Envoyer le catalogue, utiliser Envoyer le catalogue*
14. *Si l'utilisateur sélectionne Exécuter le rapport des ventes, utiliser Exécuter le rapport des ventes*
Fin Si
15. *L'utilisateur sélectionne une fonction.*
Fin de boucle
16. *Le cas d'utilisation prend fin*

TD2- Diagrammes de classe

Exercice 1 : Propriétés d'une classe

Une personne est caractérisée par son nom, son prénom, son sexe et son âge. Les objets de classe Personne doivent pouvoir calculer leurs revenus et leurs charges. Les attributs de la classe sont privés ; le nom, le prénom ainsi que l'âge de la personne doivent être accessibles par des opérations publiques.

Question : Donnez une représentation UML de la classe Personne, en remplissant tous les compartiments adéquats.

Deux types de revenus sont envisagés : d'une part le salaire et d'autre part toutes les autres sources de revenus. Les deux revenus sont représentés par des nombres réels (float). Pour calculer les charges globales, on applique un coefficient fixe de 20% sur les salaires et un coefficient de 15% sur les autres revenus.

Question : Enrichissez la représentation précédente pour prendre en compte ces nouveaux éléments.

Un objet de la classe Personne peut être créé à partir du nom et de la date de naissance. Il est possible de changer le prénom d'une personne. Par ailleurs, le calcul des charges ne se fait pas de la même manière lorsque la personne décède.

Question : Enrichissez encore la représentation précédente pour prendre en compte ces nouveaux éléments.

Exercice 2 : Relations entre classes

Question : Pour chacun des énoncés suivants, donnez un diagramme des classes :

- Tout écrivain a écrit au moins une oeuvre
- Les personnes peuvent être associées à des universités en tant qu'étudiants aussi bien qu'en tant que professeurs.
- Un rectangle a deux sommets qui sont des points. On construit un rectangle à partir des coordonnées de deux points. Il est possible de calculer sa surface et son périmètre, ou encore de le translater.
- Les cinémas sont composés de plusieurs salles. Les films sont projetés dans des salles. Les projections correspondantes ont lieu à chacune à une heure déterminée.¹
- Tous les jours, le facteur distribue des recommandés dans une zone géographique qui lui est affectée. Les habitants sont aussi associés à une zone géographique. Les recommandés sont de deux sortes : lettres ou colis. Comme plusieurs facteurs peuvent intervenir sur la même zone, on souhaite, pour chaque recommandé, le facteur qui l'a distribué, en plus du destinataire.

I

Exercice 3 : Elaboration d'un diagramme de classes complet

Un hôtel est composé d'au moins deux chambres. Chaque chambre dispose d'une salle d'eau : douche ou bien baignoire. Un hôtel héberge des personnes. Il peut employer du personnel et il est impérativement dirigé par un directeur. On ne connaît que le nom et le prénom des employés, des directeurs et des occupants. Certaines personnes sont des enfants et d'autres des adultes (faire travailler des enfants est interdit). Un hôtel a les caractéristiques suivantes : une adresse, un nombre de pièces et une catégorie. Une chambre est caractérisée par le nombre et de lits qu'elle contient, son prix et son numéro. On veut pouvoir savoir qui occupe quelle chambre à quelle date. Pour chaque jour de l'année, on veut pouvoir calculer le loyer de chaque chambre en fonction de son prix et de son occupation (le loyer est nul si la chambre est inoccupée). La somme de ces loyers permet de calculer le chiffre d'affaires de l'hôtel entre deux dates.

Question : Donnez un diagramme de classes pour modéliser le problème de l'hôtel.

Exercice 4 : Correspondance entre code Java et diagramme de classes

Question : Donnez un diagramme de classes correspondant au code source suivant :

```
public interface Dessinable {
    public void dessiner ( );
    public void effacer ( );
}
abstract public class Figure implements Dessinable {
    protected String couleur;
    protected String getCouleur ( ) { return couleur; }
    protected void setCouleur ( String c ) { couleur = c; }
}
public class Point {
    private float x;
    private float y;
    public float getX ( ) { return x; }
    public float getY ( ) { return y; }
    public void Point ( float x, float y) { ... }
}
```

```

public class Cercle extends Figure {
    private float rayon;
    private Point centre;
    public Cercle ( Point centre, float rayon) { ... }
    public void dessiner ( ) { ... }
    public void effacer ( ) { ... }
}
public class Rectangle extends Figure {
    protected Point sommets[] = new Point[2];
    public Rectangle ( Point p1, Point p2) { ... }
    public void dessiner ( ) { ... }
    public void effacer ( ) { ... }
}
public class Losange extends Figure {
    protected Point sommets[] = new Point[2];
    public Losange ( Point p1, Point p2) { ... }
    public void dessiner ( ) { ... }
    public void effacer ( ) { ... }
}

```

Exercice 5 : Héritage multiple

Question : Les étudiants et les enseignants sont deux sortes de personnes. Proposez un modèle de classes correspondant.

Question : Un doctorant est un étudiant qui assure des enseignements. Complétez le modèle de classes précédent.

Question : Les doctorants et les étudiants doivent s'inscrire au début de l'année et éventuellement modifier leur inscription. On connaît le nom et le prénom de toutes les personnes. On doit pouvoir calculer le salaire des doctorants aussi bien que celui des enseignants. Ajoutez ces éléments au modèle précédent.

Exercice 6 : Organisation des classes

Une banque compte plusieurs agences réparties sur le territoire français. Une banque est caractérisée par le nom de son directeur général, son capital global, son propre nom et de l'adresse de son siège social. Le directeur général est identifié par son nom, son prénom et son revenu. Une agence a un numéro d'agence et une adresse. Chaque agence emploie plusieurs employés, qui se caractérisent par leurs nom, prénom et date d'embauche. Les employés peuvent demander leur mutation d'une agence à une autre, mais un employé ne peut travailler que dans une seule agence. Les employés d'une agence ne font que gérer des clients. Un client ne peut avoir des comptes que dans une seule agence de la banque. Chaque nouveau client se voit systématiquement attribuer un employé de l'agence (conseiller). Les clients ont un nom, un prénom et une adresse. Les comptes sont de nature différente selon qu'ils soient rémunérés ou non (comptes courants). Les comptes rémunérés ont un taux d'intérêt et rapportent des intérêts versés annuellement.

Une première lecture de l'énoncé permettrait de faire apparaître les classes suivantes, avec leurs propriétés :

Banque	Directeur	Employé	Client
-nomDirecteur : String -capital : int -adresseSiege : String +getNomDirecteur() : String +setNomDirecteur(String n) +getCapital():int +setCapital(int capital) +getAdresseSiege():String +setAdresseSiege(String s) Banque(String Adresse)	-nom : String -prenom : String -revenu : float +getNom() : String +setNom (String n) +getPrenom ():String +setPrenom(String p) +getRevenu():float +setRevenu(float s)	-nom : String -prenom : String -dateEmbauche : Date +getNom: String +setNom(String n) +getPrenom():String +setPrenom(String s) +getDate(): Date +setDate(Date s) mutation(Agence g): boolean	-nom : String -prenom : String -adresse : String -conseiller : Employer -agence : Agence -comptes : [1..N] Compte +getNom: String +setNom(String n) +getPrenom():String +setPrenom(String s) +getDate(): Date +setDate(Dates) mutation(Agence g):boolean
CompteNonRémunéré -solde : float -numero : int ...	Agence -nomAgence :String -adresseAgence : String +getNomAgence() : String +setNomAgence(String n)	CompteRémunéré -solde : float -numero : int -taux : float ... verserInteret() : void	

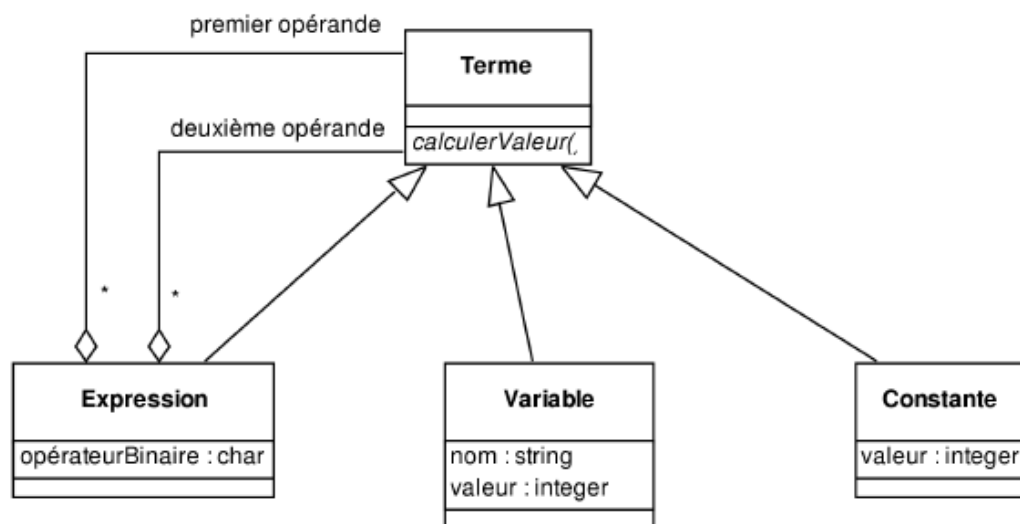
Question : Analysez ces classes et utilisez la généralisation pour factoriser au mieux la description des propriétés.

Question : Une relation particulière lie l'agence, le client, l'employé et le compte. De quelle relation s'agit-il? Donnez un diagramme de classes pour la modéliser.

Question : Donnez le diagramme de classes en n'utilisant que le nom des classes et ajoutez toutes les décorations adéquates aux associations.

Exercice 7 : Composition

Le diagramme des classes suivant permet de modéliser des expressions arithmétiques avec des opérateurs binaires comme $1+x$, $2-4$ ou encore $2-(y*4)$.



Question : Modifiez ce diagramme pour prendre en compte des opérateurs binaires (comme ici) mais aussi unaires (comme $-x$ ou $5!$ par exemple). Les expressions unaires ne doivent compter qu'un seul opérande.

TD3- Diagrammes d'activité

Exercice 1 : La cafetière

Construire un diagramme d'activité représentant l'utilisation d'une cafetière électrique:

- ✚ premier état: chercher du café
- ✚ dernier état: Servir du café

Exercice 2 : Commande d'un produit

Construire un diagramme d'activité pour modéliser le processus de commander d'un produit. Le processus concerne les acteurs suivants:

- ✚ Client: qui commande un produit et qui paie la facture
- ✚ Caisse: qui encaisse l'argent du client
- ✚ Vente: qui s'occupe de traiter et de facturer la commande du client
- ✚ Entrepôt: qui est responsable de sortir les articles et d'expédier la commande.

Exercice 3

Le logiciel de gestion des réparations est destiné en priorité au chef d'atelier, il devra lui permettre de saisir les fiches de réparations et le travail effectué par les divers employés de l'atelier.

Pour effectuer leur travail, les mécaniciens et autres employés de l'atelier vont chercher des pièces de rechange au magasin. Lorsque le logiciel sera installé, les magasiniers ne fourniront des pièces que pour les véhicules pour lesquels une fiche de réparation est ouverte; ils saisiront directement les pièces fournies depuis un terminal installé au magasin.

Lorsqu'une réparation est terminée, le chef d'atelier va essayer la voiture. Si tout est en ordre, il met la voiture sur le parc clientèle et bouclera la fiche de réparation informatisée. Les fiches de réparations bouclées par le chef d'atelier devront pouvoir être importées par le comptable dans le logiciel comptable.

1. Elaborer le diagramme d'activité pour tout le traitement de la réparation
2. Créer un diagramme d'activité pour le cas « créer une fiche de réparation »

Pour créer une fiche de réparation, le chef d'atelier saisit les critères de recherche de voitures dans le système. Le logiciel de gestion des réparations lui donne la liste des voitures correspondant aux critères entrés. Si la voiture existe, le chef d'atelier va sélectionner la voiture. Le logiciel va, ensuite, fournir les informations sur le véhicule. Si la voiture est sous garantie, le chef devra saisir la date de demande de réparation. Si la voiture n'existe pas, le chef va saisir les informations concernant ce nouveau véhicule. Dans tous les cas, le chef d'atelier devra saisir la date de réception et de restitution. Si le dommage de la voiture est payé par l'assurance, le logiciel va fournir une liste d'assurances au chef d'atelier. Ce dernier sélectionnera l'assurance adéquate. Enfin, le logiciel enregistre la fiche de réparation.

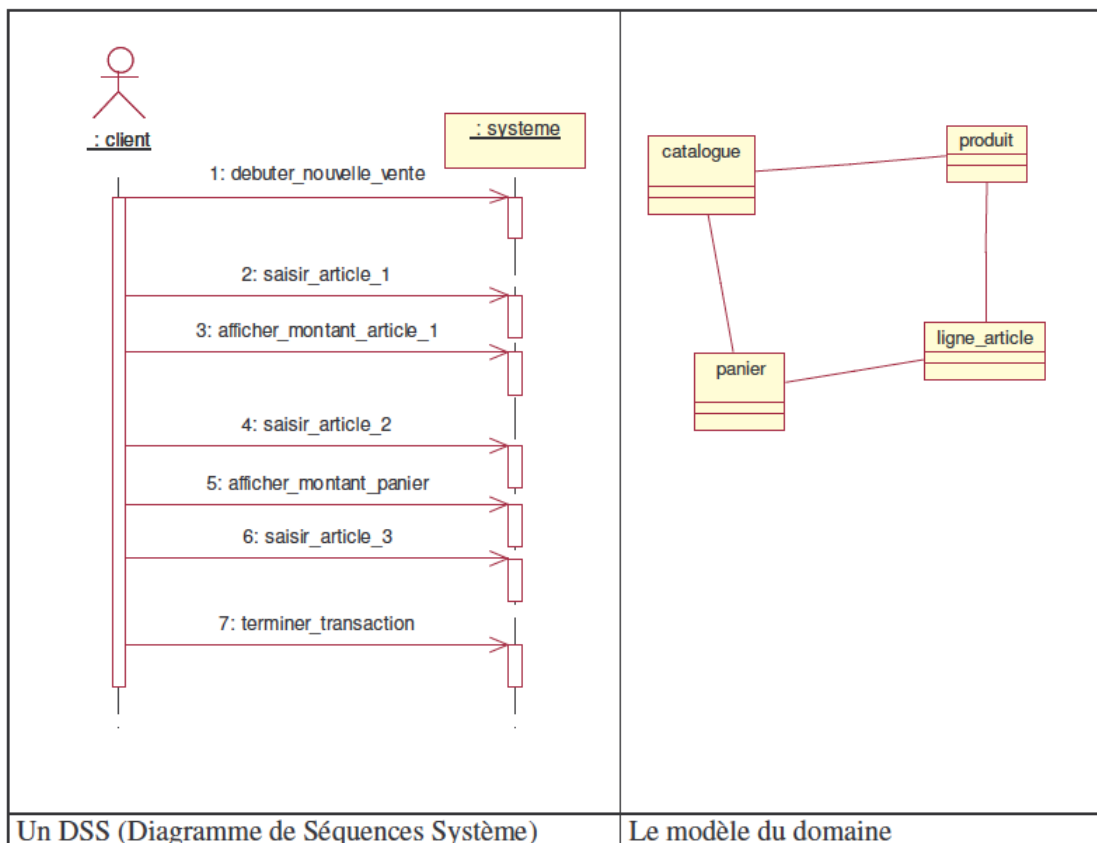
TD4- Diagrammes de séquence système et de conception

Exercice 1

Enoncé

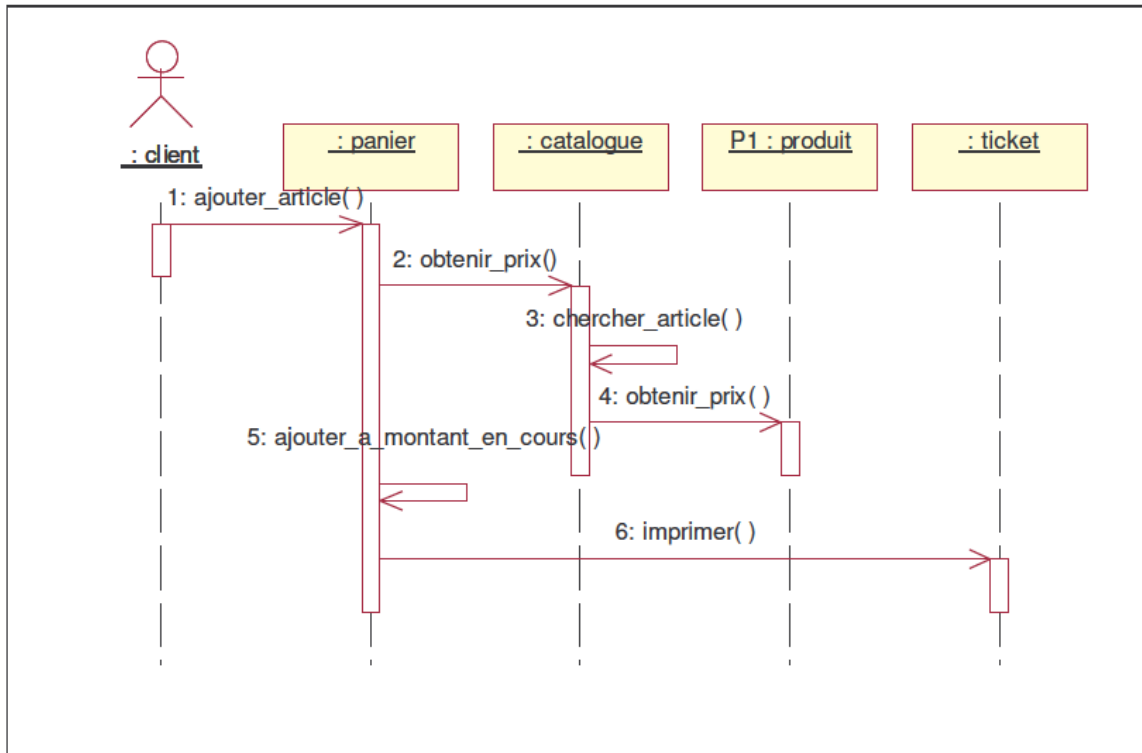
Un de vos collègues a effectué la partie analyse du projet de développement « minik's ». Il vous a remis un DSS (Diagramme de Séquences Système) et une ébauche du modèle du domaine.

Le métier étudié dans le cadre de « minik's » est celui des boutiques (Très Petits Commerces) qui disposerai à terme d'une caisse sur un ordinateur commun (PC Compatible) et s'affranchirait ainsi des point de ventes propriétaire actuellement répandus.



Questions

- Développer le diagramme de séquences système en un diagramme de séquences de conception, qui fasse apparaître tout ou partie des classes identifiées dans le modèle d'analyse (d'éventuelles classes non présentes dans le modèle du domaine peuvent être ajoutées). Cette démarche devrait permettre d'identifier une bonne partie des opérations de chaque classe.
- Mettre à jour le modèle de domaine pour refléter les dernières évolutions.



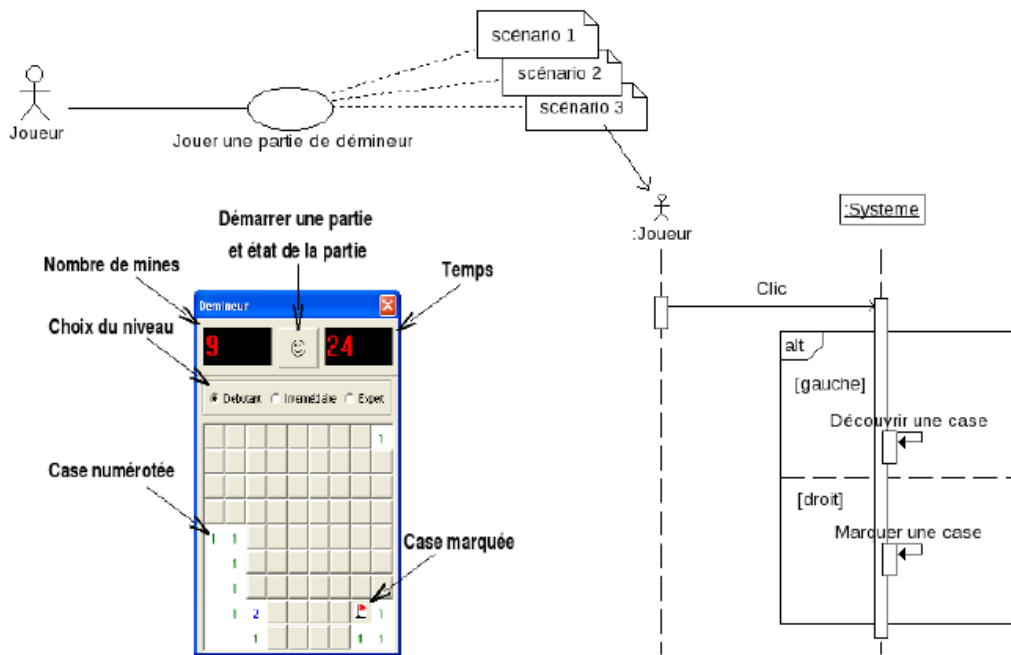
Exercice 2

Exemple

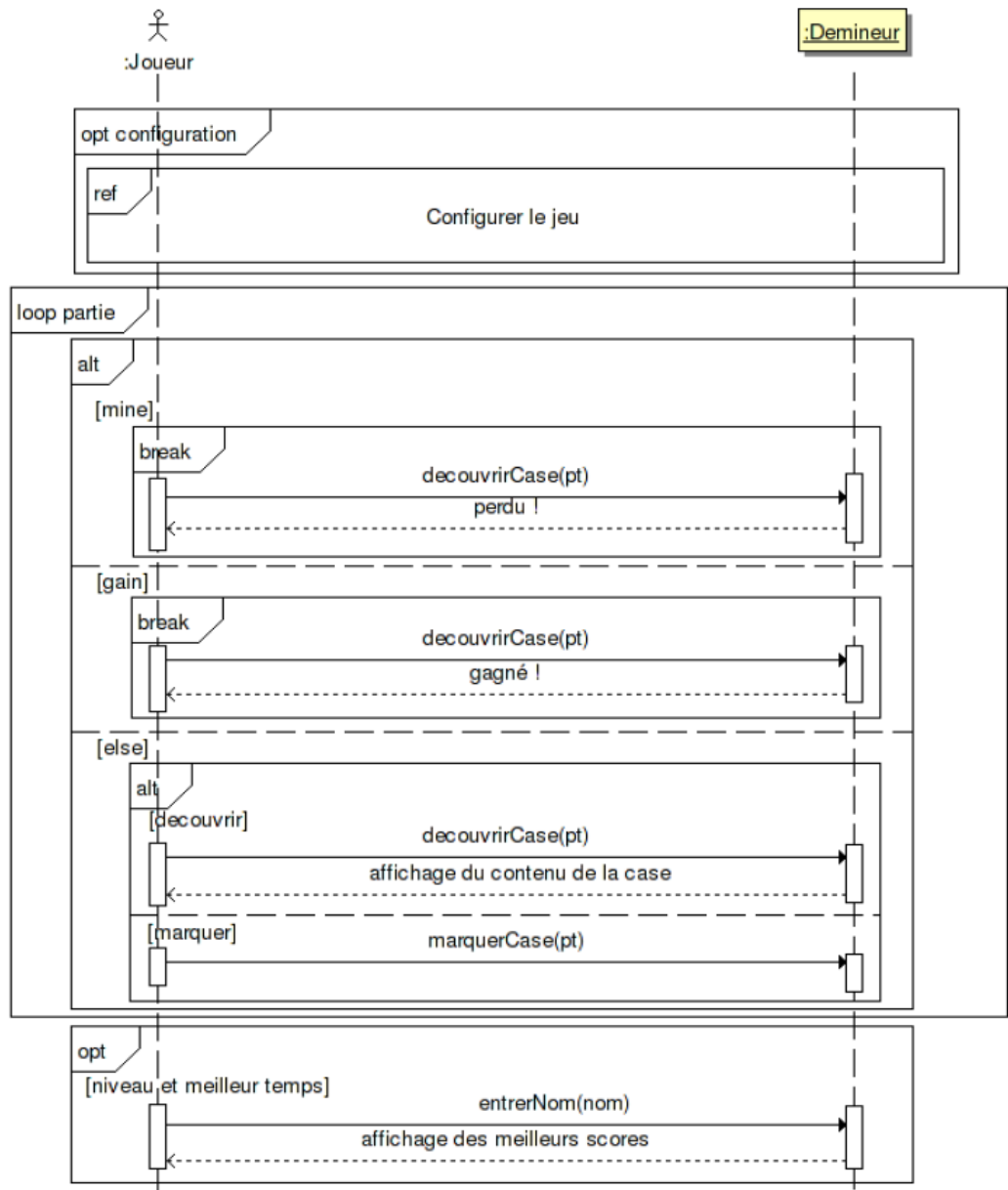
Cas d'utilisation principal : Jouer une partie de démineur

Le joueur passe son temps à découvrir ou (**alt**) marquer des cases. Dans la boucle (**loop**) de jeu, il y a trois possibilités (**alt**) : perte, gain ou cas normal. Les cas de perte ou de gain arrêtent la partie (**break**). En cas de gain, il pourra (**opt**) entrer son nom dans les meilleurs scores si ([]) il a le meilleur temps en fonction du niveau choisi.

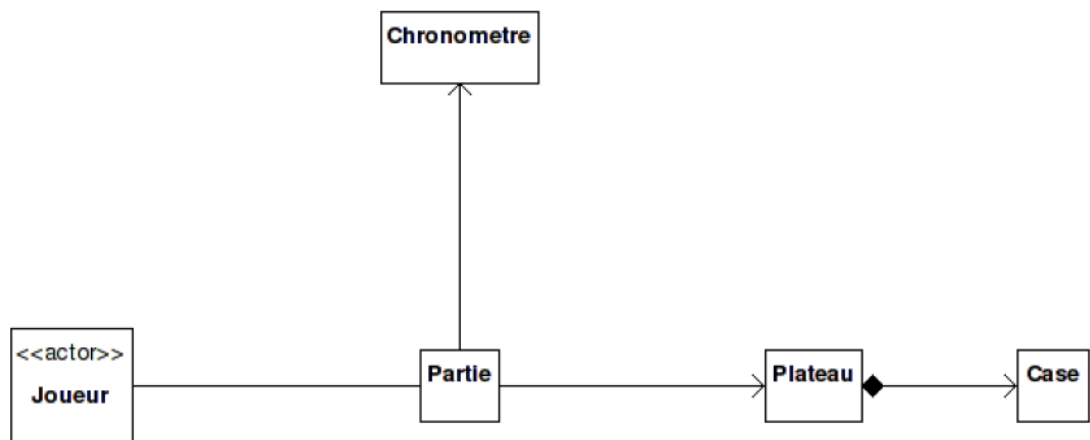
On peut ajouter la configuration du jeu comme étape optionnelle (**opt**) avant de commencer à jouer.



1. Développer plus en détail le diagramme de séquence système.



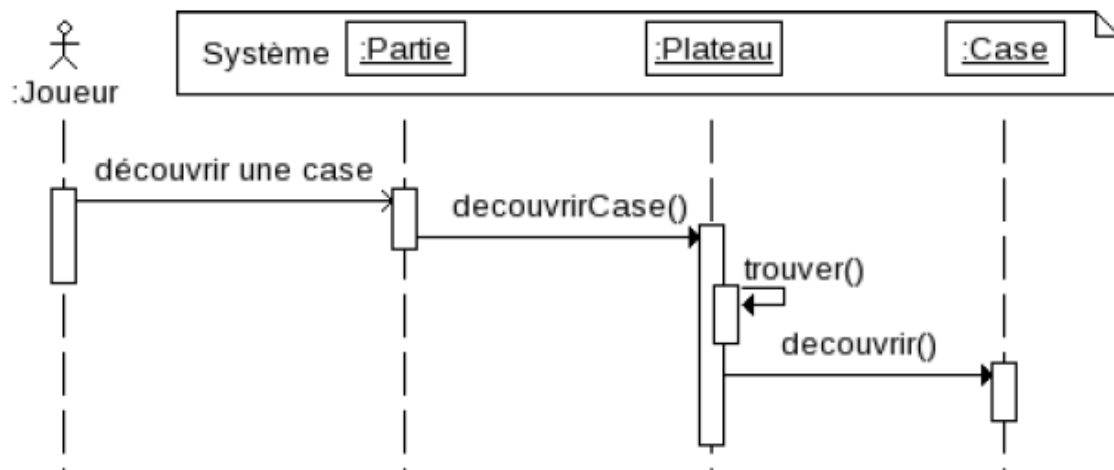
2. Construire un premier jet de diagramme de classe



3. Elaborer le diagramme de séquence de conception selon la consigne suivante

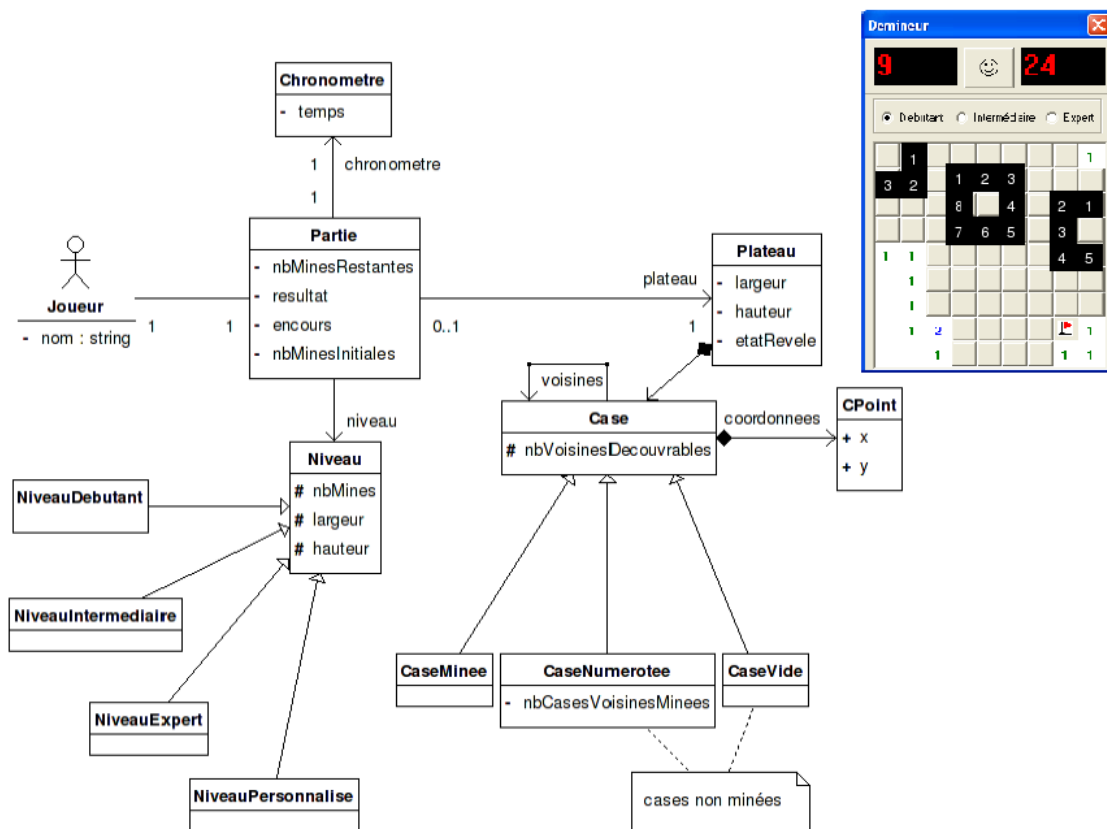
Passer de l'analyse à la conception

Ensuite, on remplacera le système vu comme une « boîte noire » (du point de vue de l'analyse) par des objets logiciels (du point de vue de la conception).



4. Compléter le diagramme de classe élaboré en 2. En sachant que :

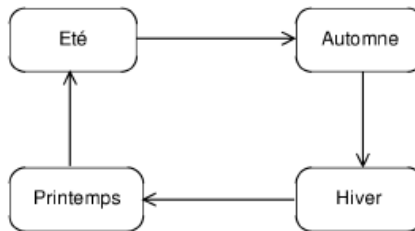
Une analyse plus profonde du domaine nous permet de spécialiser les types de **Case**. Le **Plateau** est défini pas une hauteur et une largeur (suivant le **Niveau** sélectionné par le joueur) et les cases se doivent d'être repérables par un système de coordonnées (x,y). Une case possède 3, 5 ou 8 voisins.



TD5- Diagrammes d'états

Exercice 1 : Transitions « change » et « after »

Le changement de saisons constitue une boucle continue. On considère un objet de la classe « Saison » de durée de vie infinie. En utilisant des événements de type « change » ou « after », donner le diagramme des états-transitions de la classe Saison correspondant aux états de l'année climatique de la France (printemps, été, automne, hiver).



Exercice 2 : Etats associés à une classe

Considérons une classe Partie dont la responsabilité est de gérer le déroulement d'une partie de jeu d'échecs. Cette classe peut être dans deux états :

- le tour des blancs
- le tour des noirs.

Les événements à prendre en considération sont

- un déplacement de pièces de la part du joueur noir
- un déplacement de pièces de la part du joueur blanc
- la demande de prise en compte d'un échec et mat par un joueur. S'il est validé par la classe partie, un échec et mat assure la victoire du dernier joueur. Dans ce cas, une activité « noirsGagnants » ou « blancsGagnants » – selon le cas – est déclenchée (appel de méthode).
- la demande de prise en compte d'un pat qui mène aussi à une fin de partie, avec une égalité. Dans ce cas, une activité « égalité » est déclenchée.

Donner le diagramme d'états-transitions associé à la classe Partie

Exercice 3 : Etats composites et états composés

Une montre digitale propose une fonction horloge et une fonction chronomètre. Elle est munie de quatre boutons, l'appui sur chacun des boutons donnant lieu à un événement :

- Le bouton « light », quand il est pressé, allume une lumière. La lumière reste allumée pendant une durée de deux secondes, après quoi elle s'éteint à nouveau. Un appui sur ce bouton alors que la lumière réinitialise la durée l'allumage de la lumière.
- Le bouton « mode » active successivement les trois modes principaux de la montre : l'affichage de l'heure courante, le mode chronomètre et le mode réglage (pour mettre à jour l'heure courante).
- En mode chronomètre, le bouton « start/stop » lance et arrête le chronomètre. En mode réglage, il active successivement les fonctions de réglage de l'heure, des minutes ou des secondes.
- En mode réglage, le bouton « set » incrémente les heures, les minutes ou les secondes. En mode chronomètre, il remet le chronomètre à zéro.

Nous allons décomposer l'étude du fonctionnement de cette montre en plusieurs étapes, en établissement des diagrammes d'états/transition concernant la classe montre.

3.1 Etats composites

L'affichage est géré par un objet ayant le rôle d'« afficheur » et disposant d'une opération afficher qui permet d'afficher la chaîne de caractères qu'on lui donne en paramètre. Le décompte du temps est géré par un objet ayant le rôle de « time » et disposant d'une opération toString retournant une chaîne de caractères correspondant à l'heure courante.

Question : Modélisez dans un premier temps le comportement lié à l'utilisation du bouton « mode » de la montre.

Question : Dans le même diagramme, ajoutez le comportement lié à l'utilisation du bouton « light ».

■

Question : Donnez aussi un diagramme de classes pour modéliser les relations et les opérations des objets de type « Afficheur », « Time » et « Montre ».

3.2 Etats composés

En mode chronomètre, le bouton « start-stop » lance ou arrête le chronomètre. Le bouton « set » remet le chronomètre à zéro s'il est arrêté.

Question : Modélisez ces comportements.

Il est en fait possible de lancer le chronomètre, puis de basculer en mode affichage de l'heure et de revenir au mode chronomètre. Dans ce cas, le chronomètre continue de tourner pendant cette opération.

Question : Ajoutez cette possibilité de retrouver l'état précédent en changeant de mode. Vous pourrez utiliser un pseudo-état historique.

Le décompte du temps chronométré est géré par un objet avec le rôle de « chrono » et qui dispose des opérations stop, start, init et enfin toString retournant une chaîne de caractères.

Question : Ajoutez une classe « Chronomètre » au diagramme de classes.

L'afficheur dispose de méthodes `indicateurHeure`, `indicateurMinute` et `indicateurSeconde` pour afficher l'indicateur adéquat. La classe « Time » dispose de méthodes `incrementeHeure`, `incrementeMinute` et `initSeconde`.

En mode réglage, le bouton start-stop active successivement les fonctions de réglage de l'heure, des minutes ou des secondes. Selon la fonction de réglage courante, l'afficheur affiche un indicateur sous les heures, les minutes ou les secondes. Le bouton set incrémente l'heure courante d'une heure en mode

réglage de l'heure, d'une minute en mode réglage des minutes, ou remet les secondes à zéro en mode réglage des secondes. L'heure affichée ne change plus dès qu'on entre dans le mode réglage.

Question : Modélisez ces comportements à l'aide d'un diagramme d'états-transitions.

Question : Enrichissez encore le diagramme de classes de manière à ce qu'il soit cohérent avec le diagramme d'états/transitions.